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

Yuanzhi Li (CMU), Colin Wei (Stanford), Tengyu Ma (Stanford)

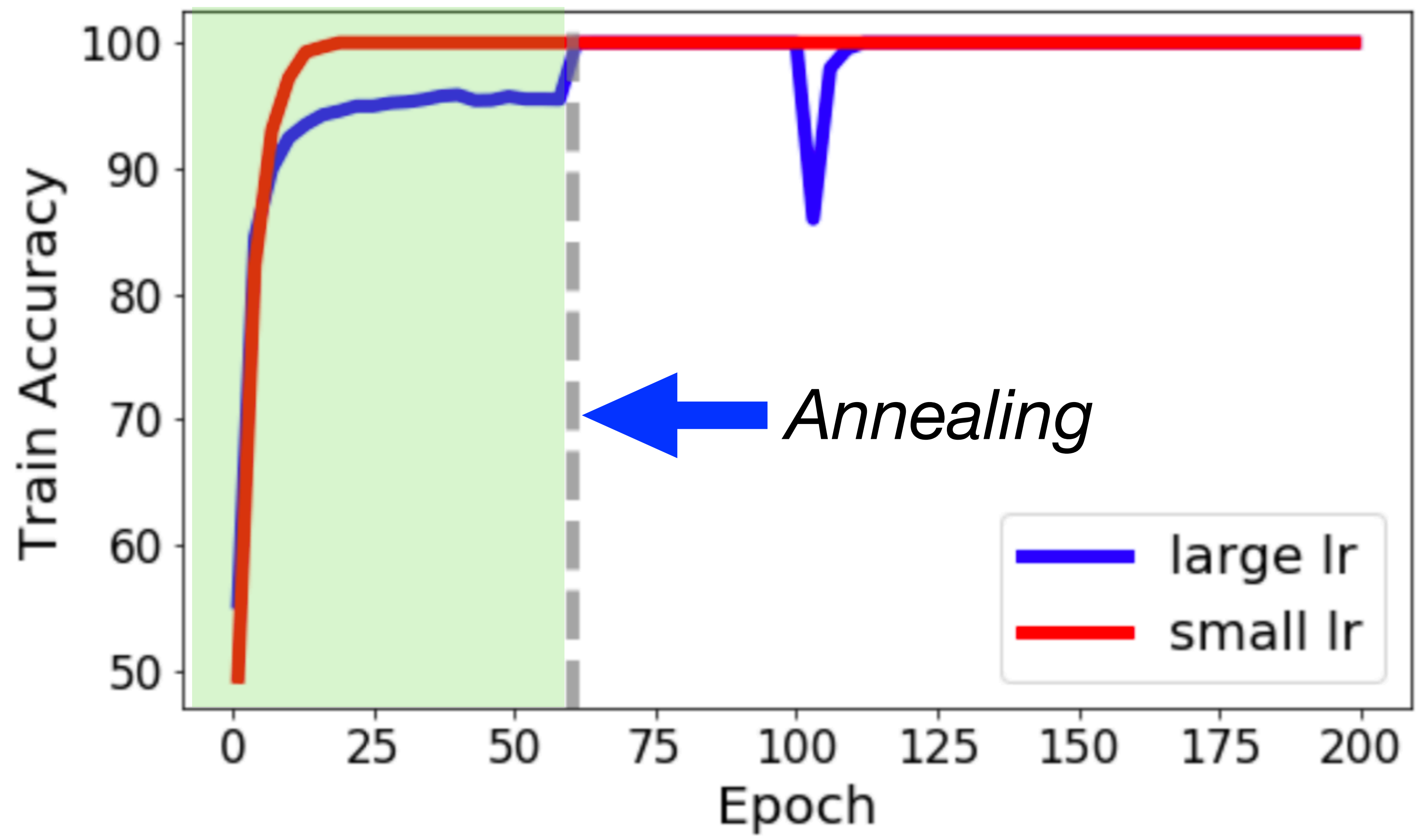
NeurIPS 2019

Presenter: Jiao, Wenxiang

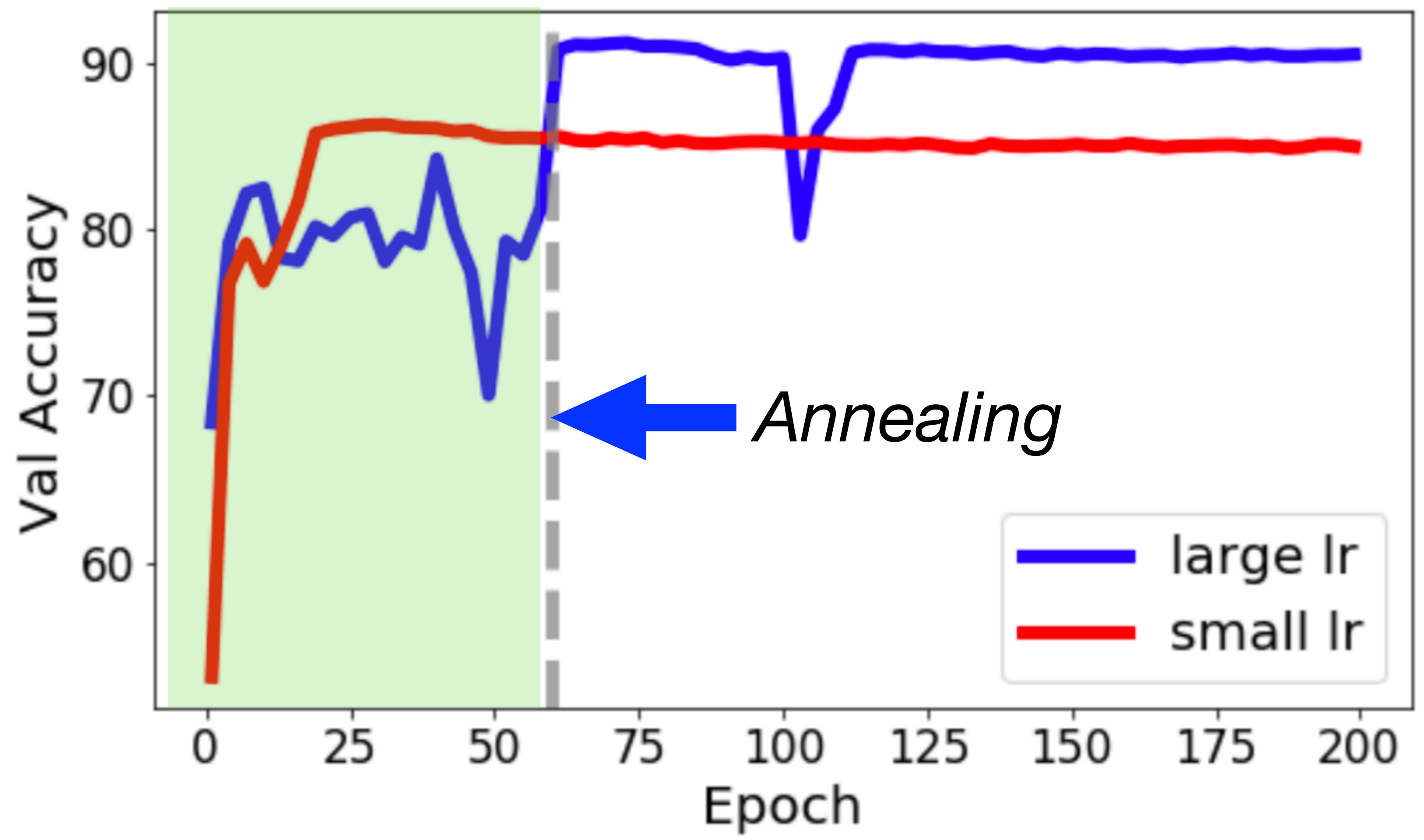
Large Initial Learning Rate is Crucial for Generalization

- Common schedule: large initial learning rate + annealing
- … But small learning rate: better train and test performance up until annealing ?

Train Accuracy



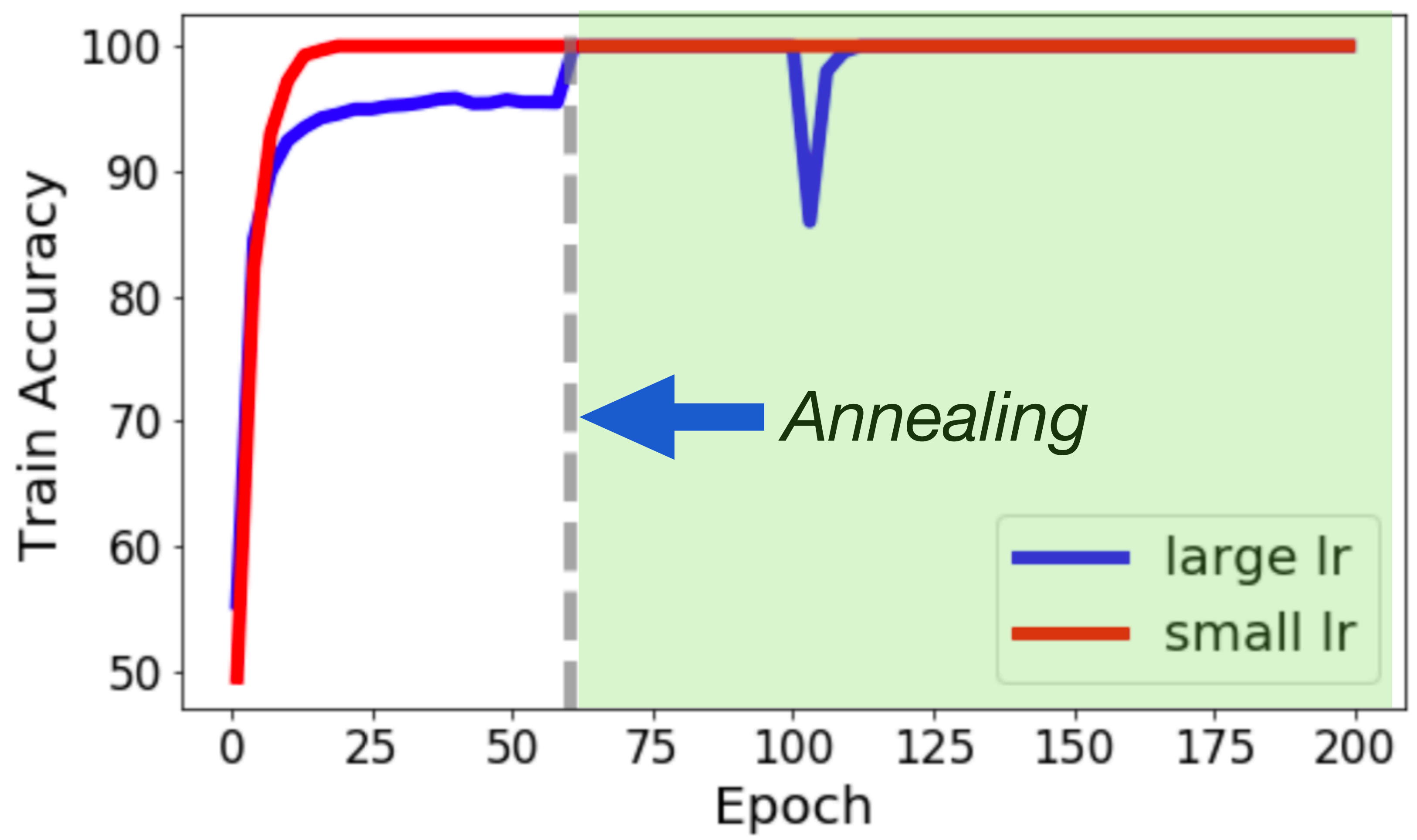
Valid Accuracy



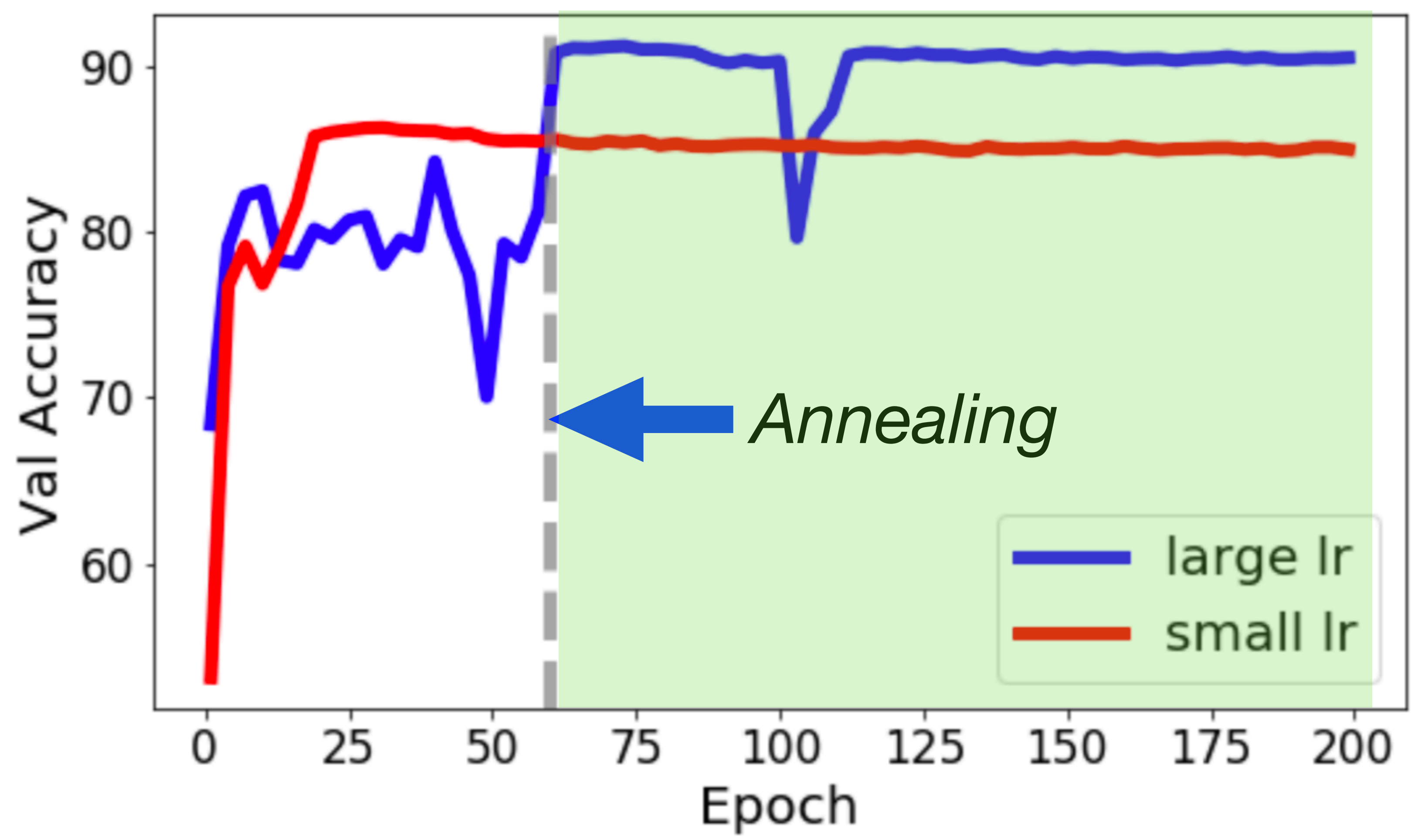
Large Initial Learning Rate is Crucial for Generalization

- Common schedule: large initial learning rate + annealing
- … But small learning rate: better train and test performance up until annealing ?

Train Accuracy



Valid Accuracy

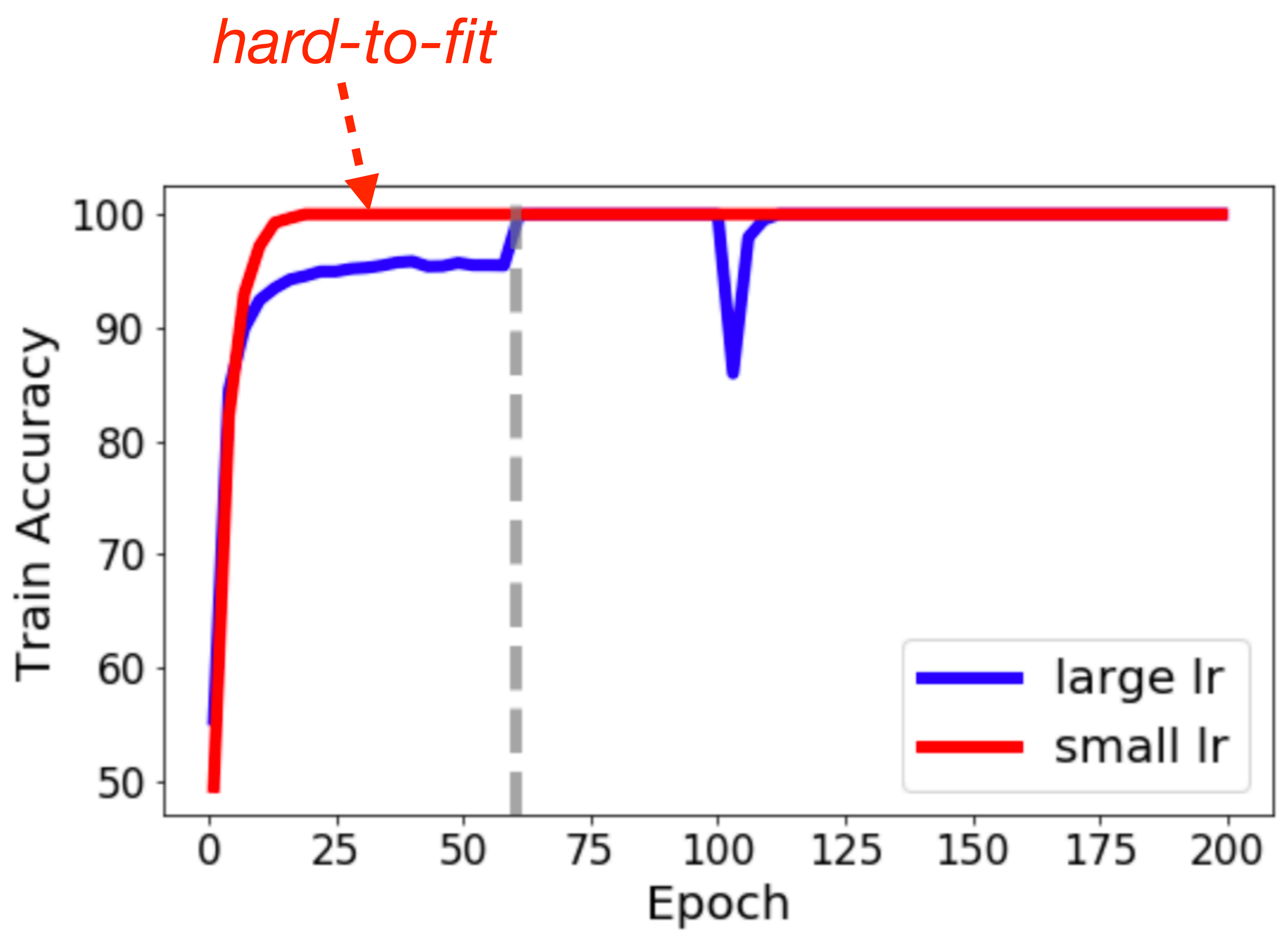


- Large LR outperforms small LR after annealing!

LR schedule changes order of learning patterns => generalization

- Small LR quickly memorizes hard-to-fit "class signatures"
  - Ignores other patterns, harming generalization

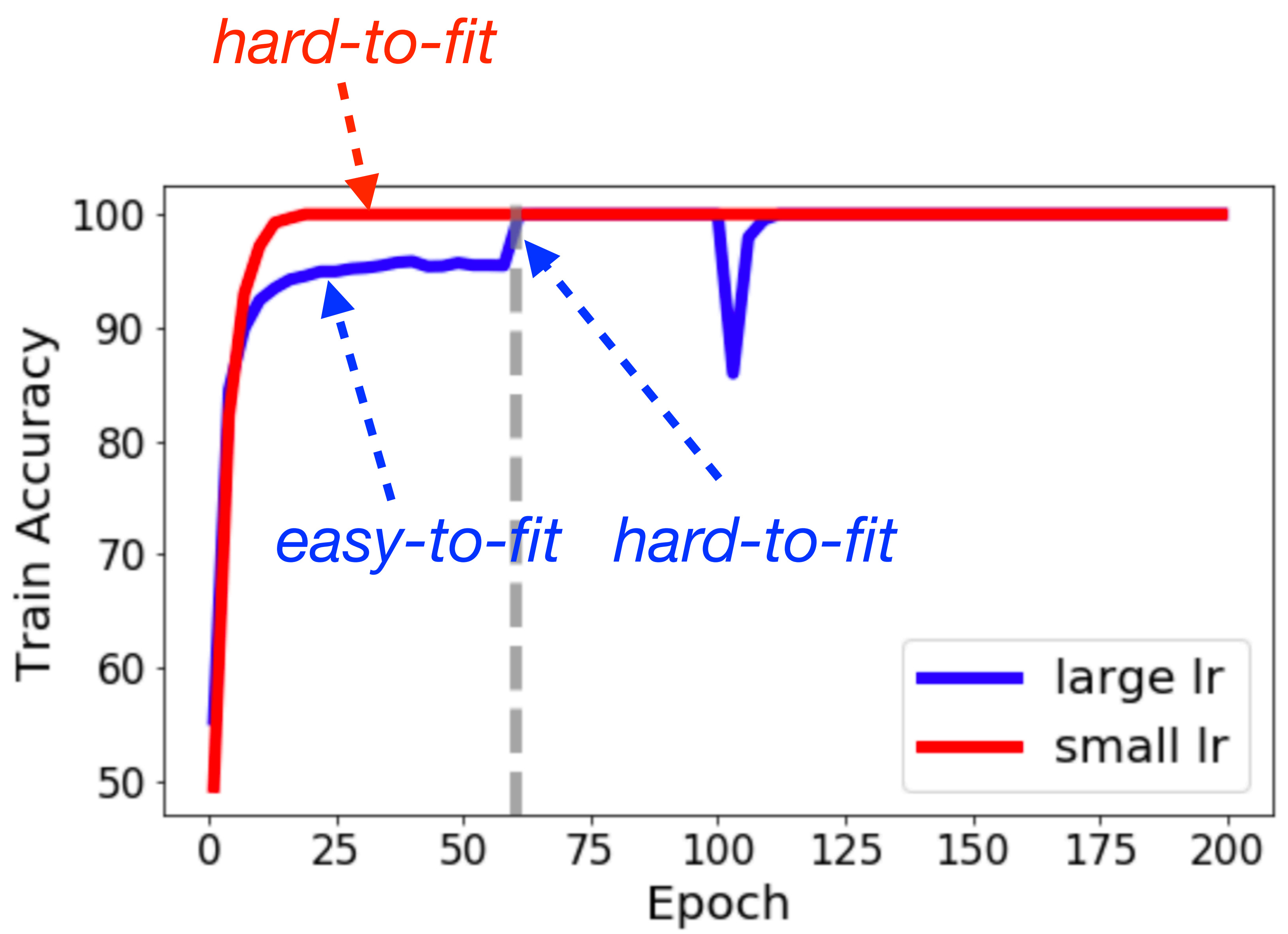
Key features indicating the corresponding class



LR schedule changes order of learning patterns => generalization

- Small LR quickly memorizes hard-to-fit “class signatures”
  - Ignores other patterns, harming generalization
- Large initial LR + annealing learns easy-to-fit patterns first
  - Only memorizes hard-to-fit patterns after annealing
  - => learns to use all patterns, helping generalization

Key features indicating the corresponding class



LR schedule changes order of learning patterns => generalization

- Small LR quickly memorizes hard-to-fit "class signatures"
  - Ignores other patterns, harming generalization
- Large initial LR + annealing learns easy-to-fit patterns first
  - Only memorizes hard-to-fit patterns after annealing
  - => learns to use all patterns, helping generalization

Key features indicating the corresponding class

- Intuition: large LR
  - => larger SGD noise
  - => effectively weaker representational power
  - => won't overfit to "signatures"

LR schedule changes order of learning patterns => generalization

- Small LR quickly memorizes hard-to-fit "class signatures"
  - Ignores other patterns, harming generalization
- Large initial LR + annealing learns easy-to-fit patterns first
  - Only memorizes hard-to-fit patterns after annealing
  - => learns to use all patterns, helping generalization

Key features indicating the corresponding class

- Intuition: large LR
  - => larger SGD noise
  - => effectively weaker representational power
  - => won't overfit to "signatures"

- Non-convexity is crucial: different LR schedules find different solutions
  - For convex problems, both LR schedules find same solution

LR schedule changes order of learning patterns => generalization

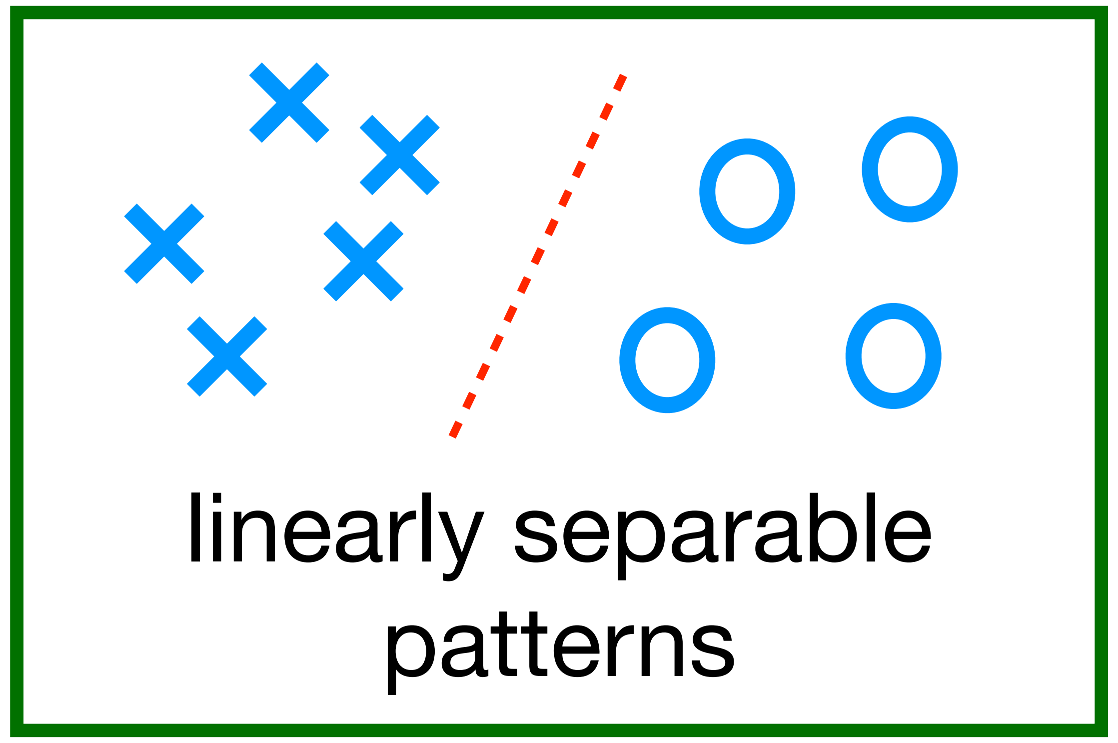
- This work: setting where LR schedule provably changes learning order, causing generalization gap

Can be proved theoretically!

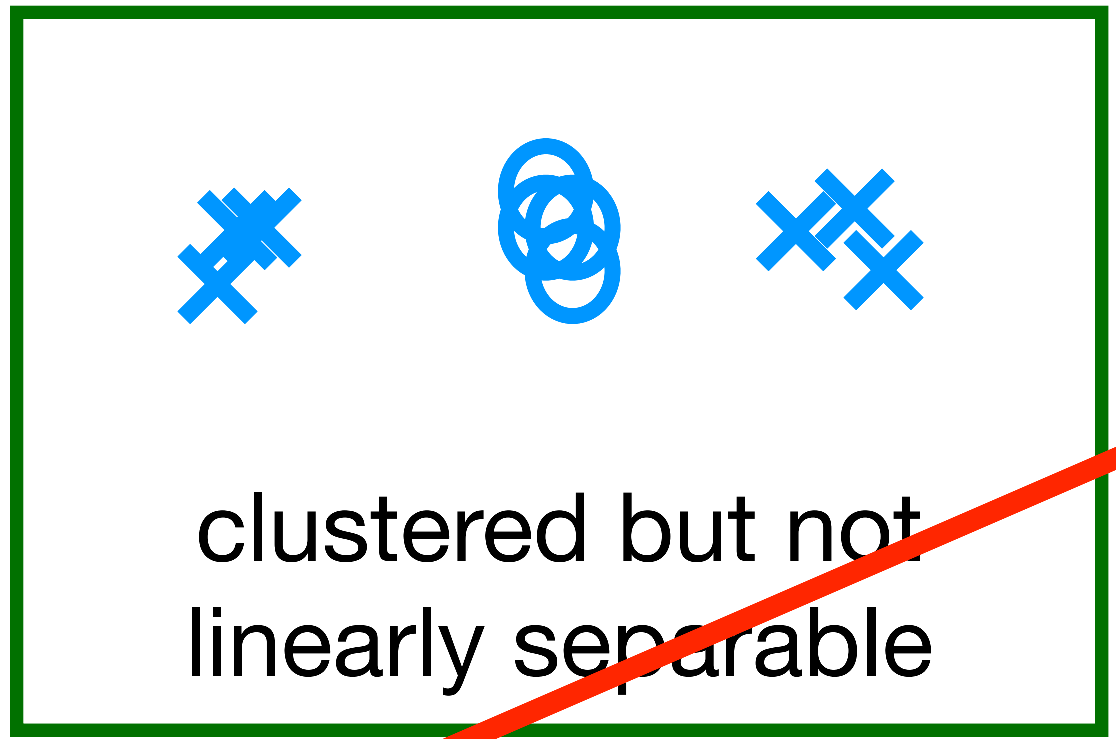
Theoretical Setting

- Three types of examples:

Group 1: 20% examples with hard-to-generalize, easy-to-fit patterns (***pattern A only***)



Group 2: 20% examples with easy-to-generalize, hard-to-fit patterns (***pattern B only***)



Group 3: 60% examples with both patterns (***pattern A and B***)

~~First *d* coordinates: pattern A~~

Last *d* coordinates: pattern B

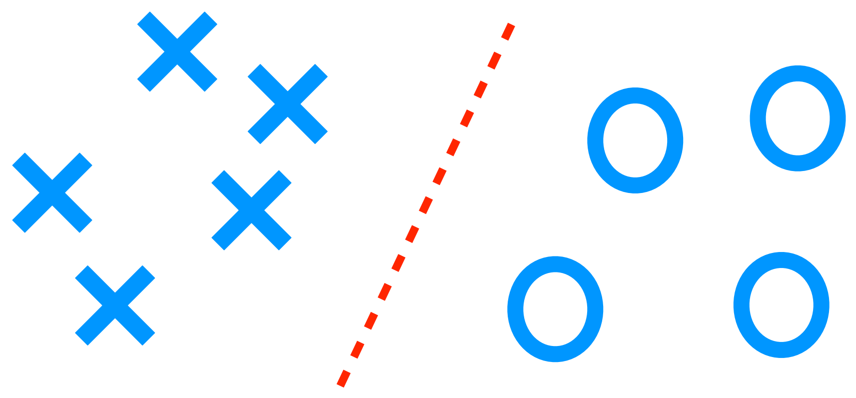
Small LR: quickly memorize pattern B, ignore pattern A from Group 3

=> Only learn pattern A from 0.2N examples in Group 1

Theoretical Setting

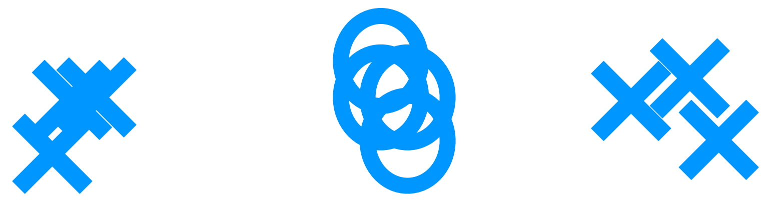
- Three types of examples:

Group 1: 20% examples with hard-to-generalize, easy-to-fit patterns (***pattern A only***)



linearly separable patterns

Group 2: 20% examples with easy-to-generalize, hard-to-fit patterns (***pattern B only***)



clustered but not linearly separable

Group 3: 60% examples with both patterns (***pattern A and B***)

First *d* coordinates: pattern A

Last *d* coordinates: pattern B

Small LR: quickly memorize pattern B, ignore pattern A from Group 3
=> Only learn pattern A from 0.2N examples in Group 1

Large LR: first learn pattern A, SGD noise prevents learning pattern B until after annealing
=> Learn pattern A from 0.8N total examples!

Theorems

Theorem 1.1 (Informal, large initial LR + anneal)**.** *There is a dataset with size N of the form (1.1) such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially only learn hard-to-generalize, easy-to-fit patterns from the $0.8N$ examples containing such patterns.

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ and still learns all easy-to-generalize, hard to fit patterns correctly with $0.2N$ samples.

Theorem 1.2 (Informal, small initial LR)**.** *In the same setting as above, with small initial learning rate the network will:*

1) quickly learn all easy-to-generalize, hard-to-fit patterns.

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ examples containing both pattern types, and only learn them from the $0.2N$ examples containing only hard-to-generalize patterns.

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ and will perform relatively worse on these patterns at test time.

Theorems

Theorem 1.1 (Informal, large initial LR + anneal)**.** *There is a dataset with size N of the form* (1.1) *such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially $\boxed{\textit{only learn hard-to-generalize, easy-to-fit}}$ *patterns from the* $0.8N$ *examples containing such patterns.*

Poor margin on pattern B before annealing (lemma 4.1-4.2)

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ *and still learns all easy-to-generalize, hard to fit patterns correctly with* $0.2N$ *samples.*

Theorem 1.2 (Informal, small initial LR)**.** *In the same setting as above, with small initial learning rate the network will:*

1) quickly learn all easy-to-generalize, hard-to-fit patterns.

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ *examples containing both pattern types, and only learn them from the* $0.2N$ *examples containing only hard-to-generalize patterns.*

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ *and will perform relatively worse on these patterns at test time.*

Theorems

Theorem 1.1 (Informal, large initial LR + anneal). *There is a dataset with size N of the form (1.1) such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially only learn hard-to-generalize, easy-to-fit patterns from the $0.8N$ examples containing such patterns.

Poor margin on pattern B before annealing (lemma 4.1-4.2)

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ and still learns all easy-to-generalize, hard to fit patterns correctly with $0.2N$ samples.

Converge fast after annealing with low final loss on both pattern A and B (lemma 4.3-4.4)

Theorem 1.2 (Informal, small initial LR). *In the same setting as above, with small initial learning rate the network will:*

1) quickly learn all easy-to-generalize, hard-to-fit patterns.

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ examples containing both pattern types, and only learn them from the $0.2N$ examples containing only hard-to-generalize patterns.

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ and will perform relatively worse on these patterns at test time.

Theorems

Theorem 1.1 (Informal, large initial LR + anneal). *There is a dataset with size N of the form (1.1) such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially $\boxed{\textit{only learn hard-to-generalize, easy-to-fit}}$ *patterns from the* $0.8N$ *examples containing such patterns.*

Poor margin on pattern B before annealing (lemma 4.1-4.2)

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ *and still* $\boxed{\textit{learns all easy-to-generalize, hard to fit}}$ *patterns correctly with* $0.2N$ *samples.*

Converge fast after annealing with low final loss on both pattern A and B (lemma 4.3-4.4)

Theorem 1.2 (Informal, small initial LR). *In the same setting as above, with small initial learning rate the network will:*

Converge to small training error too quickly (lemma 5.1)

1) $\boxed{\textit{quickly learn all easy-to-generalize, hard-to-fit}}$ *patterns.*

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ *examples containing both pattern types, and only learn them from the* $0.2N$ *examples containing only hard-to-generalize patterns.*

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ *and will perform relatively worse on these patterns at test time.*

Theorems

Theorem 1.1 (Informal, large initial LR + anneal). *There is a dataset with size N of the form (1.1) such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially only learn hard-to-generalize, easy-to-fit patterns from the $0.8N$ examples containing such patterns.

Poor margin on pattern B before annealing (lemma 4.1-4.2)

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ and still learns all easy-to-generalize, hard to fit patterns correctly with $0.2N$ samples.

Converge fast after annealing with low final loss on both pattern A and B (lemma 4.3-4.4)

Theorem 1.2 (Informal, small initial LR). *In the same setting as above, with small initial learning rate the network will:*

Converge to small training error too quickly (lemma 5.1)

1) quickly learn all easy-to-generalize, hard-to-fit patterns.

Small magnitude of gradient on Group 2+3 (lemma 5.2)

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ examples containing both pattern types, and only learn them from the $0.2N$ examples containing only hard-to-generalize patterns.

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ and will perform relatively worse on these patterns at test time.

Theorems

Theorem 1.1 (Informal, large initial LR + anneal). *There is a dataset with size N of the form (1.1) such that with a large initial learning rate and noisy gradient updates, a two layer network will:*

1) initially only learn hard-to-generalize, easy-to-fit patterns from the $0.8N$ examples containing such patterns.

Poor margin on pattern B before annealing (lemma 4.1-4.2)

2) learn easy-to-generalize, hard-to-fit patterns only after the learning rate is annealed.

Thus, the model learns hard-to-generalize, easily fit patterns with an effective sample size of $0.8N$ and still learns all easy-to-generalize, hard to fit patterns correctly with $0.2N$ samples.

Converge fast after annealing with low final loss on both pattern A and B (lemma 4.3-4.4)

Theorem 1.2 (Informal, small initial LR). *In the same setting as above, with small initial learning rate the network will:*

Converge to small training error too quickly (lemma 5.1)

1) quickly learn all easy-to-generalize, hard-to-fit patterns.

Small magnitude of gradient on Group 2+3 (lemma 5.2)

2) ignore hard-to-generalize, easily fit patterns from the $0.6N$ examples containing both pattern types, and only learn them from the $0.2N$ examples containing only hard-to-generalize patterns.

Thus, the model learns hard-to-generalize, easily fit patterns with a smaller effective sample size of $0.2N$ and will perform relatively worse on these patterns at test time.

Poor margin on pattern A (lemma 5.3)

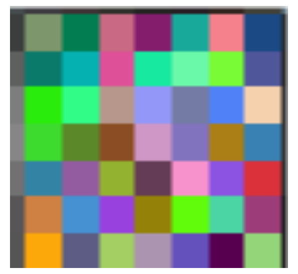
Experimental Verification

- Three types of examples: on CIFAR-10

Group 1: 20% examples with hard-to-generalize, easy-to-fit patterns (***pattern A only***)

original image

Group 2: 20% examples with easy-to-generalize, hard-to-fit patterns (***pattern B only***)



Hard-to-fit patch indicating class

Group 3: 60% examples with both patterns (***pattern A and B***)

Patch: a random vector z with i.i.d entries from Gaussian distribution for each class, with a scalar multiple

Experimental Verification

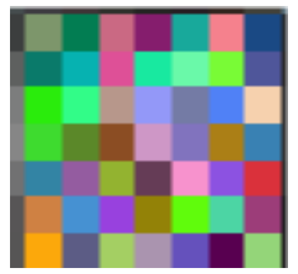
- Three types of examples: on CIFAR-10

Group 1: 20% examples with hard-to-generalize, easy-to-fit patterns (***pattern A only***)

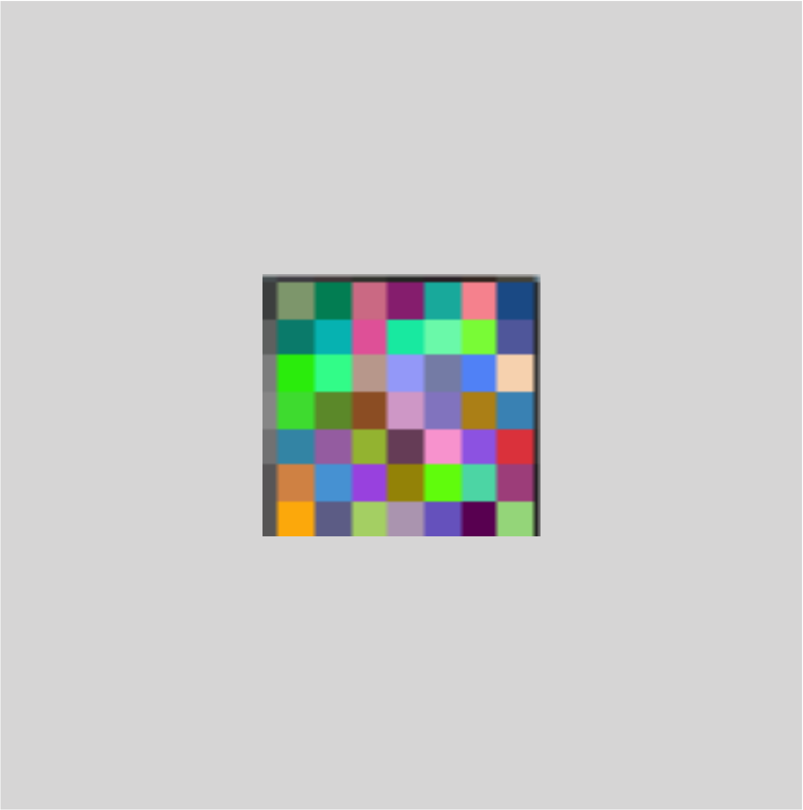
Group 2: 20% examples with easy-to-generalize, hard-to-fit patterns (***pattern B only***)

Group 3: 60% examples with both patterns (***pattern A and B***)

original image



Hard-to-fit patch indicating class



Small LR: quickly memorize pattern B, ignore pattern A from Group 3

=> Only learn pattern A from 0.2N examples in Group 1

Experimental Verification

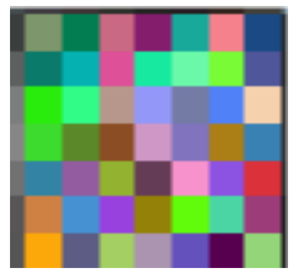
- Three types of examples: on CIFAR-10

Group 1: 20% examples with hard-to-generalize, easy-to-fit patterns (***pattern A only***)

Group 2: 20% examples with easy-to-generalize, hard-to-fit patterns (***pattern B only***)

Group 3: 60% examples with both patterns (***pattern A and B***)

original image



Hard-to-fit patch
indicating class

Large LR: first learn pattern A, SGD noise prevents learning pattern B until after annealing

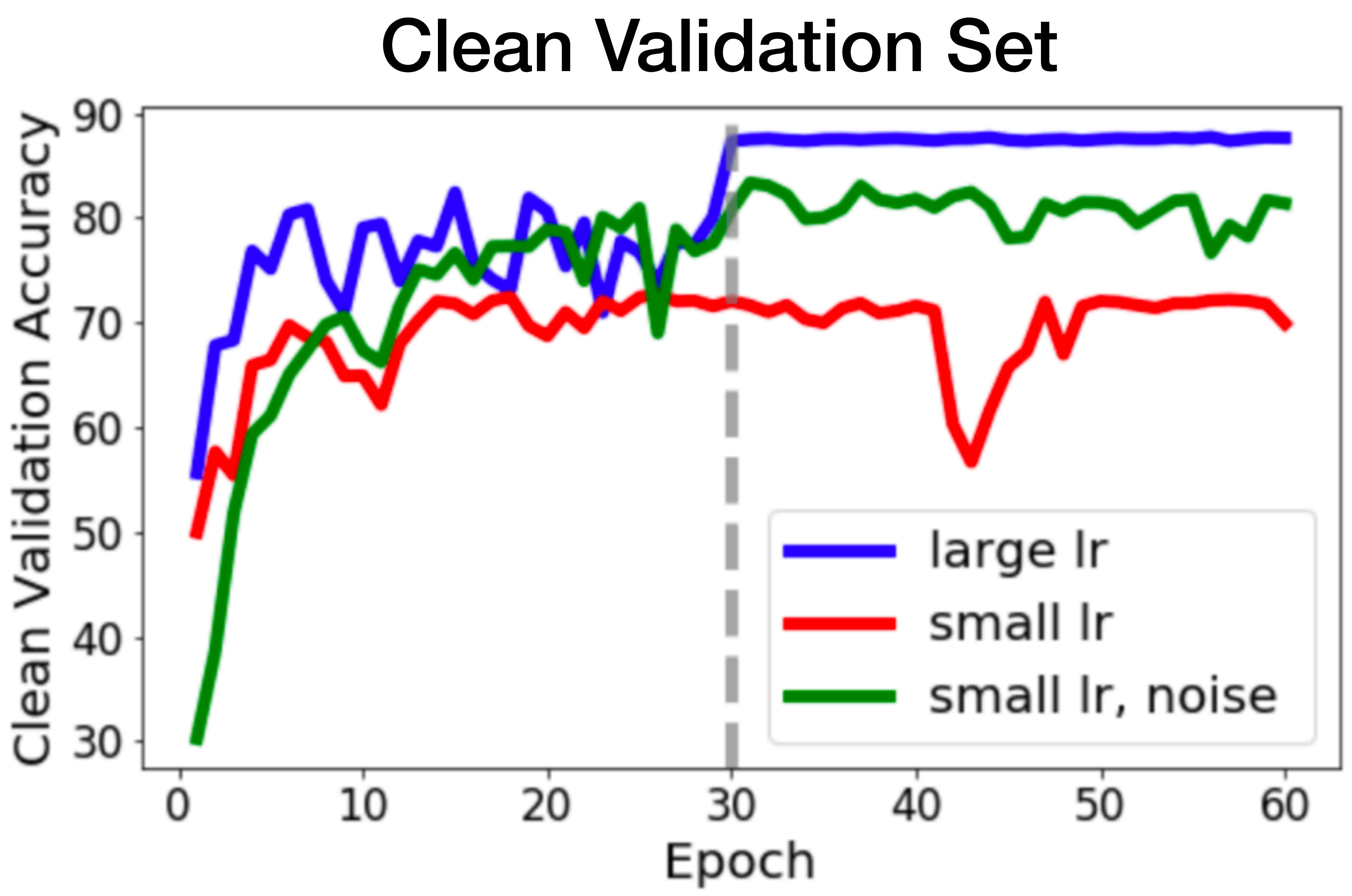
=> Learn pattern A from 0.8N total examples!

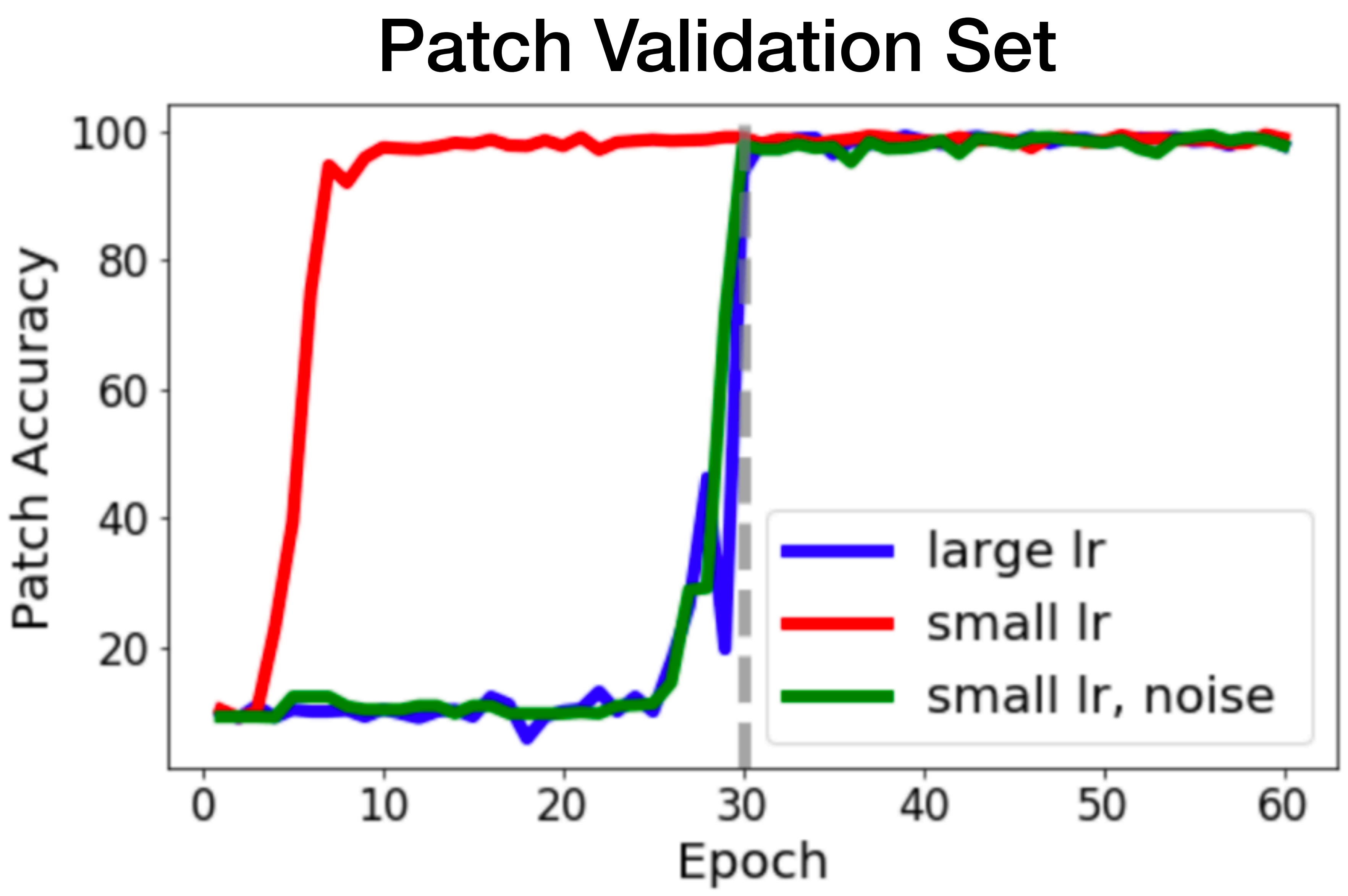
Experimental Verification

- Expected results: on CIFAR-10
  - Small LR overfits to patches quickly => higher accuracy on ***patches at beginning***
  - Small LR learns less on pattern A => lower accuracy on ***original images***

Experimental Verification

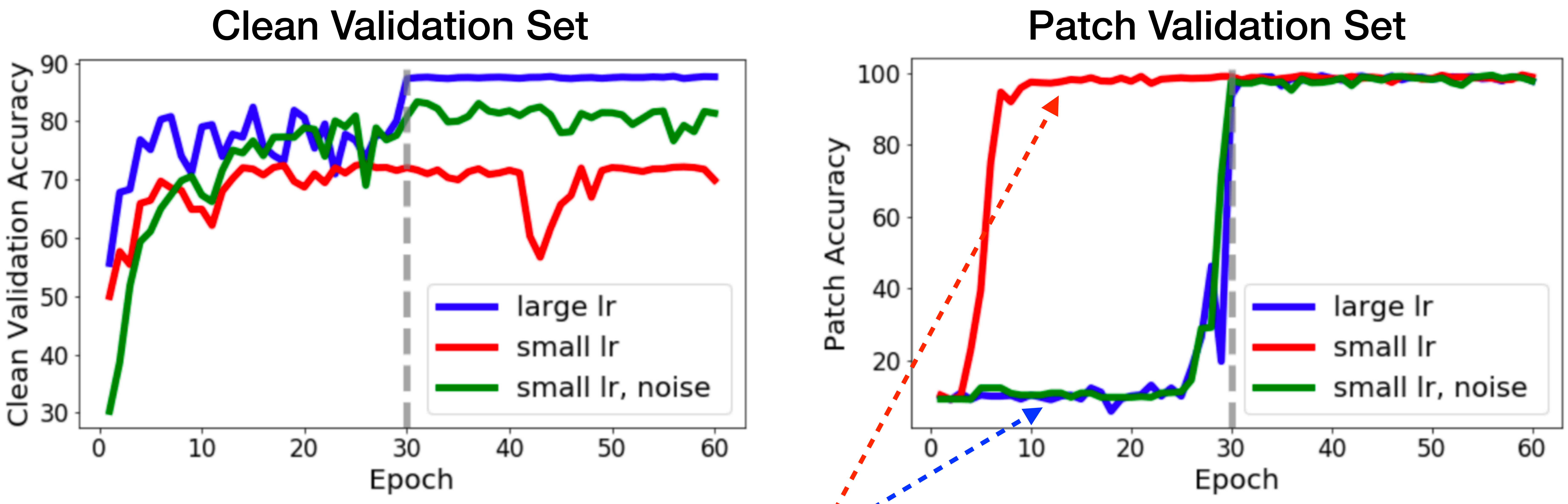
- Learning behavior: on modified CIFAR-10





Experimental Verification

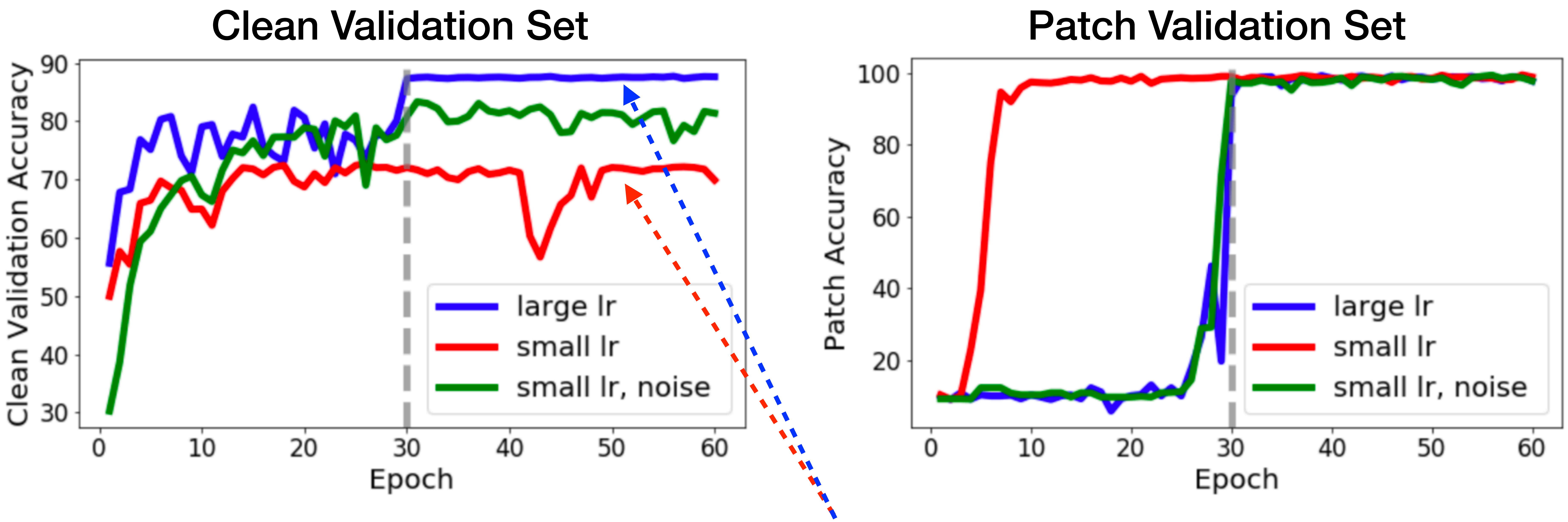
- Learning behavior: on modified CIFAR-10



Order of learning patterns does differ between the two LR schedules!

Experimental Verification

- Learning behavior: on modified CIFAR-10



Small LR learns less pattern A => worse performance

Experimental Verification

- Performance: original CIFAR-10 vs. modified CIFAR-10

Method	Val. Acc
Large LR + anneal	90.41%
Small LR + noise	89.65%
Small LR	84.93%

Method	Mixed Val. Acc.	Clean Val. Acc.
Large LR + anneal	95.35%	87.61%
Small LR	92.83%	69.89%
Small LR + noise	94.43%	81.36%

Performance drop: Small LR encounters a more significant drop on the modified CIFAR-10

=> Large LR + anneal: 90.41% —> 87.61% (*-2.80%*)

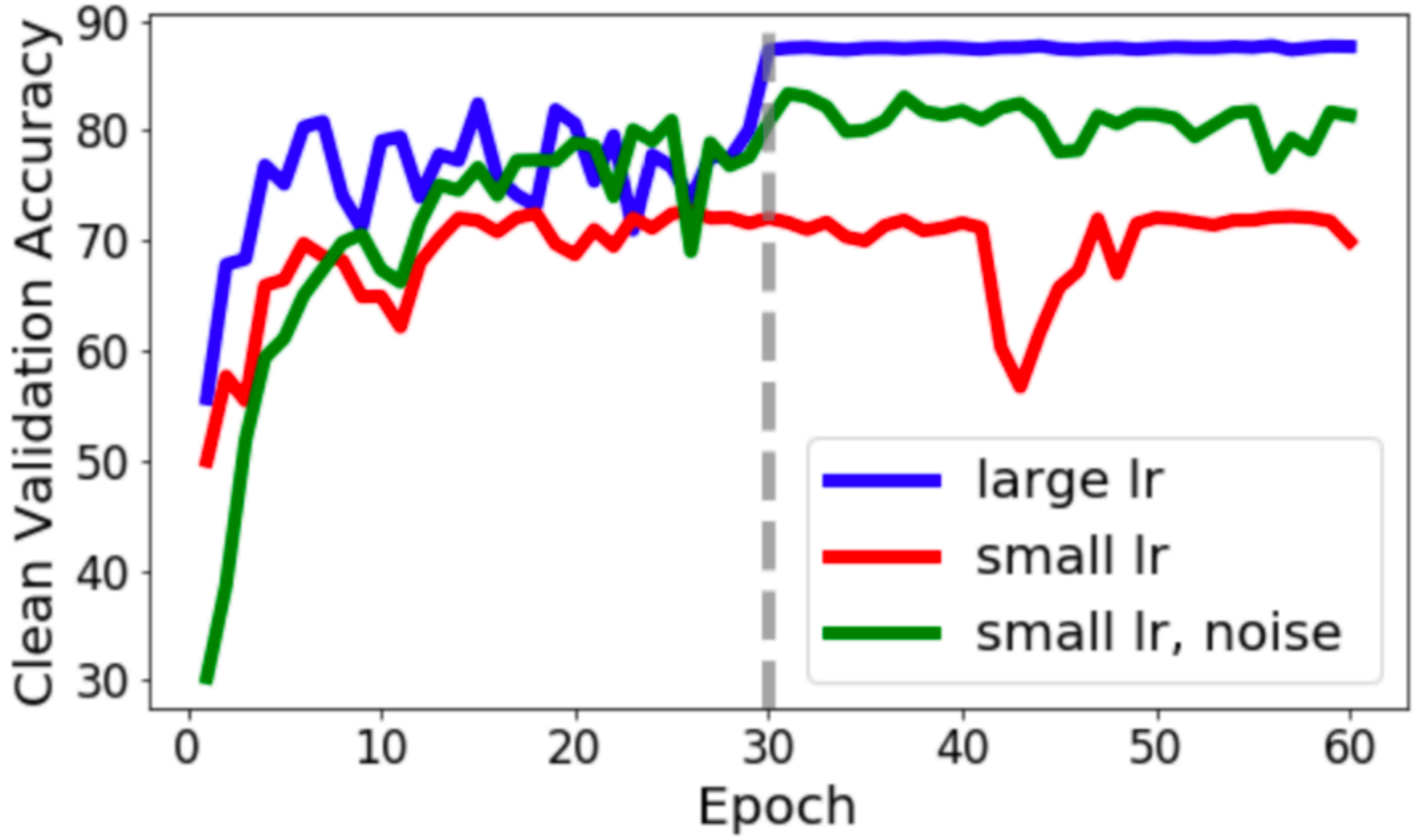
=> Small LR: 84.93% —> 69.89% (*-15.04%*)

Overfit to patch => more performance drop

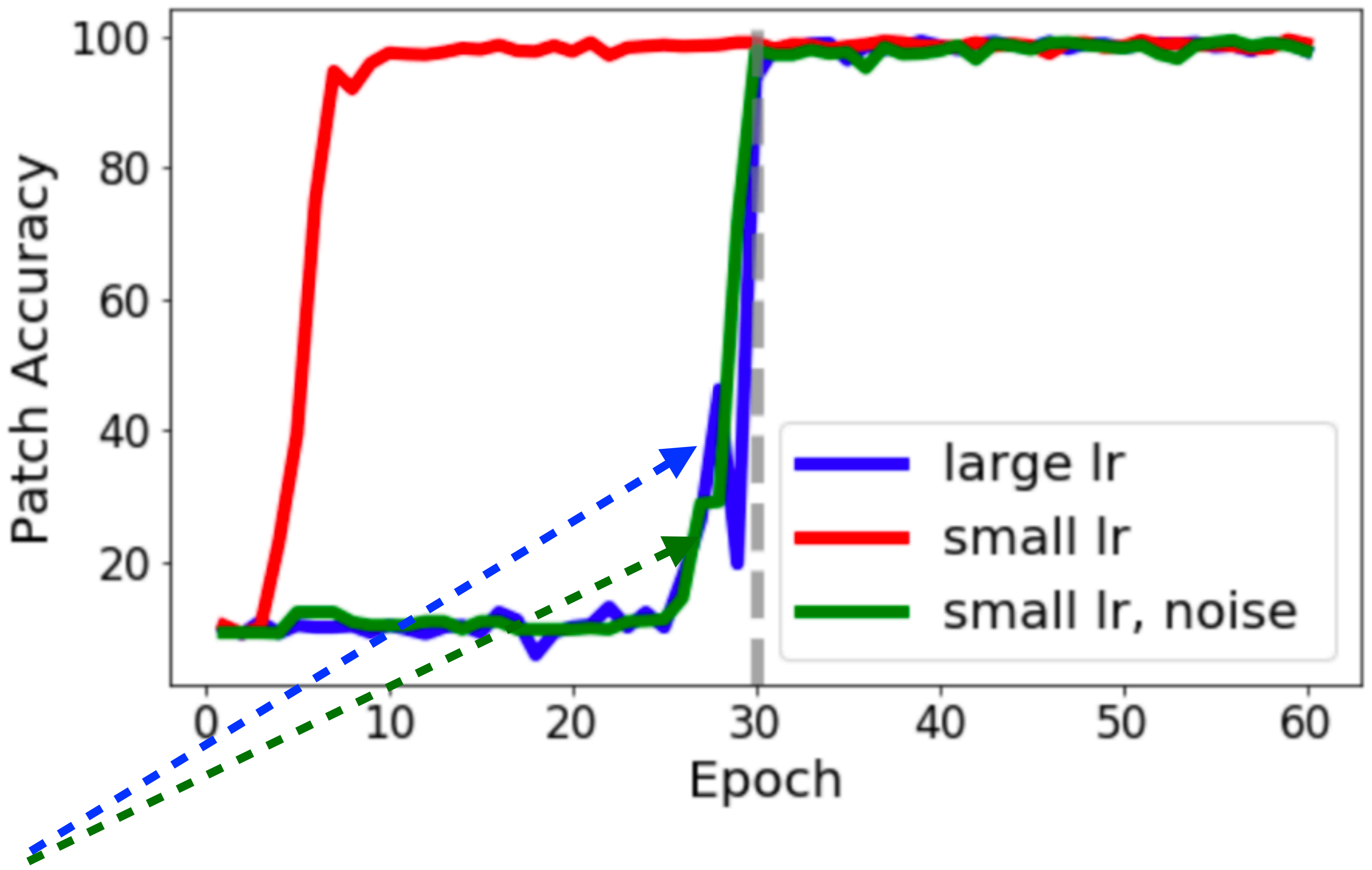
Experimental Verification

- Possible solution to Small LR
  - Large LR: large SGD noise
  - => Small LR + noise (annealed over time)

Clean Validation Set



Patch Validation Set

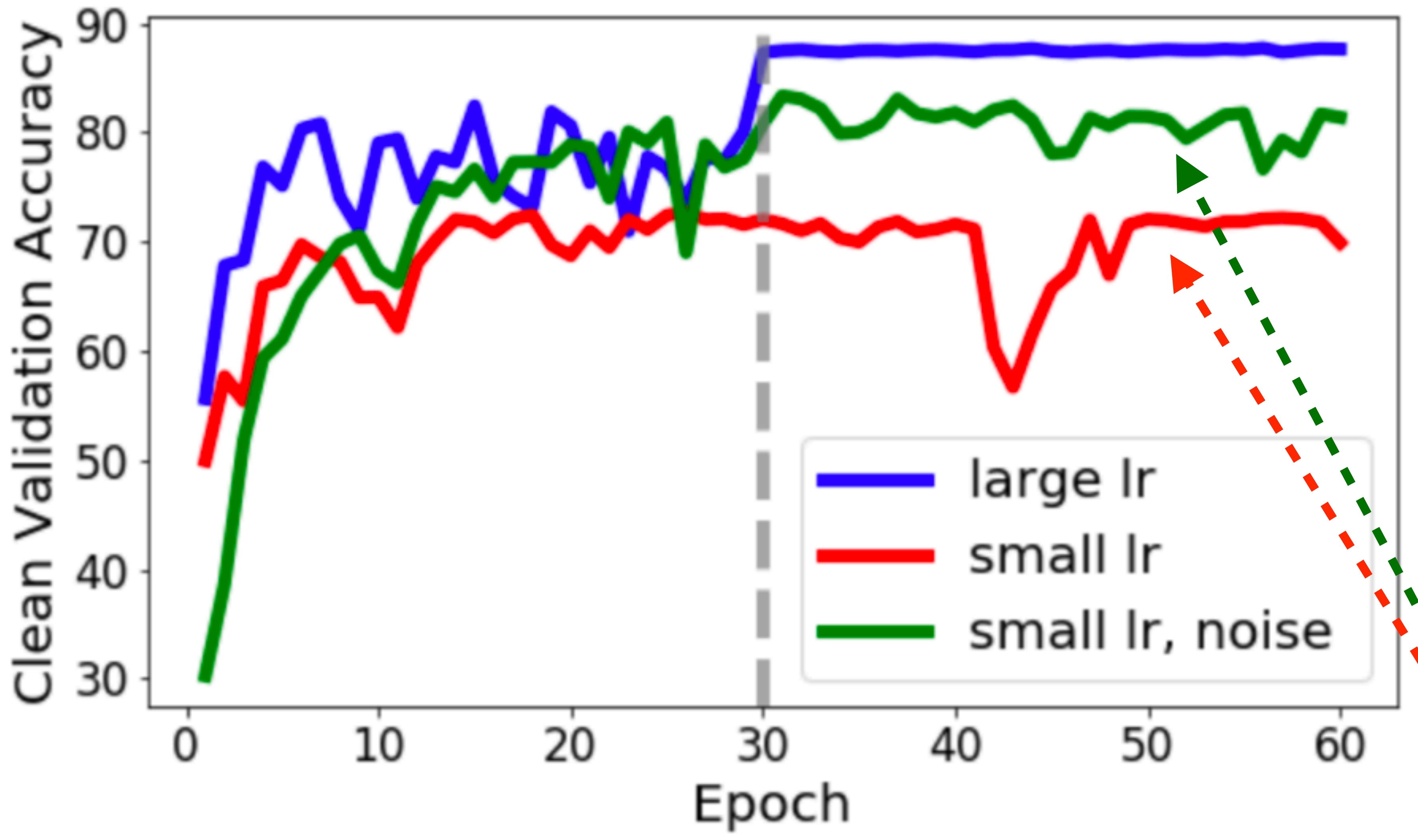


Small LR + noise shows similar behaviors as Large LR + annealing

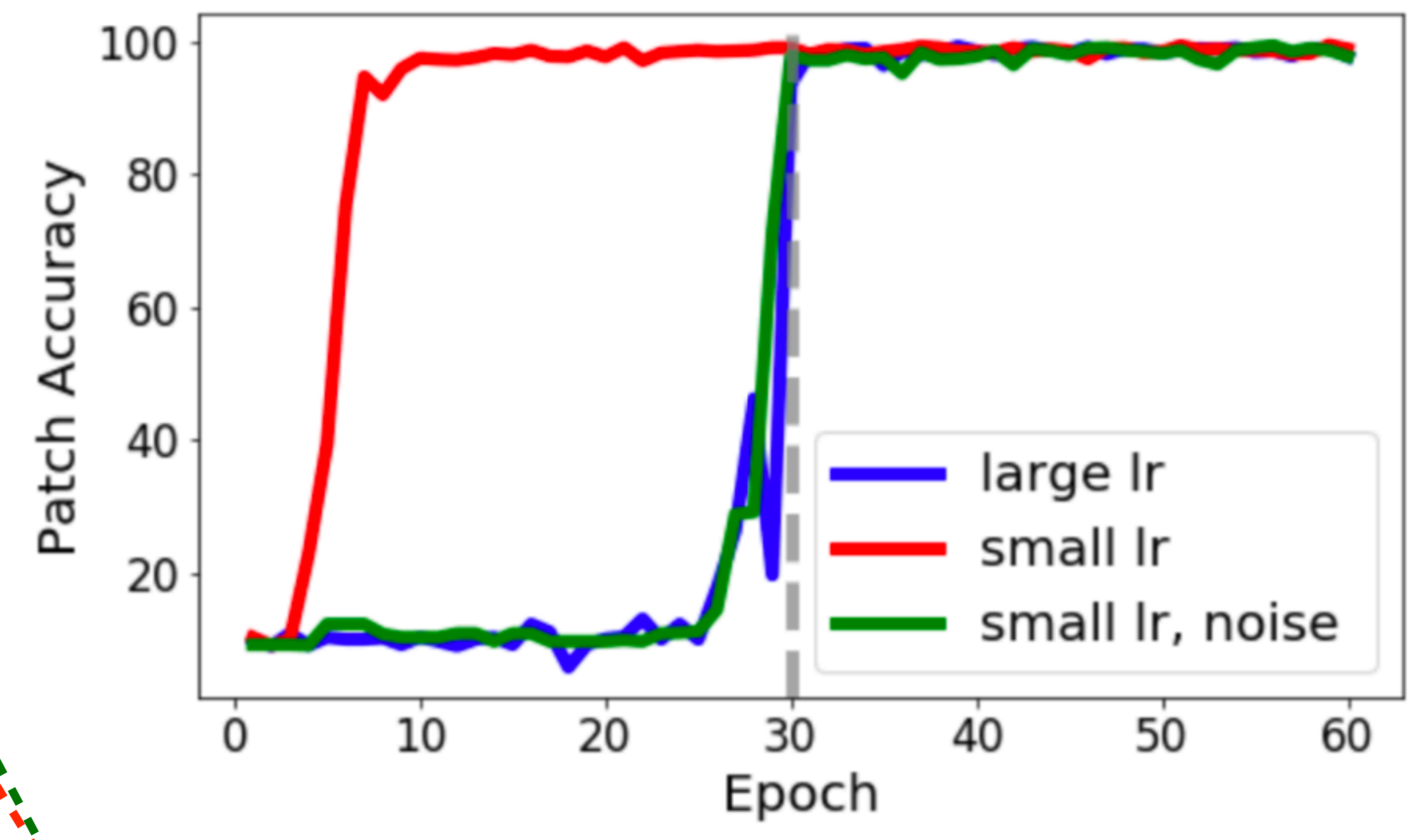
Experimental Verification

- Possible solution to Small LR
  - Large LR: large SGD noise
  - => Small LR + noise (annealed over time)

Clean Validation Set



Patch Validation Set



Small LR + noise leads to improvement

Summary

- Linking LR schedules with order of learning patterns is very interesting
- The claims are supported by theoretical proof and experimental validation
- Definition of patterns and design of experiments are inspiring