SPEECH COMMANDS

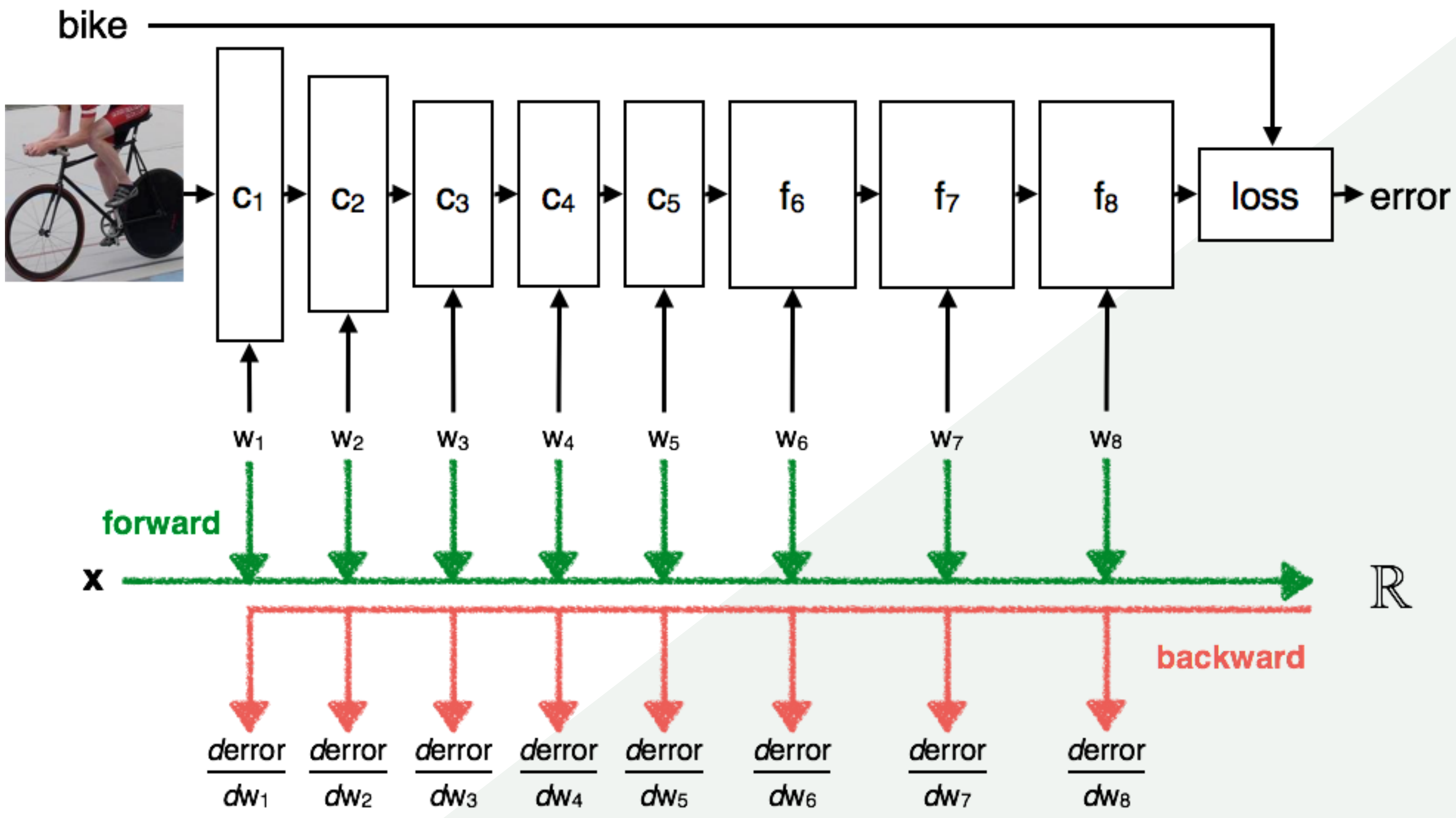
TEAM 4

Presented by Somayeh, Juan Carlos & Guillem

PROBLEM STATEMENT

- 30 English words
- Recorded by the contribution of several thousands persons
- The collected dataset includes 65,000 one-second recordings

How can we seperate these recordings and put in these 30 classes?



BASELINE WITH VGG

BASELINE

With a full 30-epoch training...

ACCURACY	**95.9%**
AV. LOSS	**0.2482**

... seems difficult to improve.

Kernel Shape
Modification
5x3

JAN 26

MIXUP
Data set augmentation

JAN 27

VGG parallel to LSTM

JAN 28

Presentation

JAN 30

IMPROVEMENTS & CHANGES

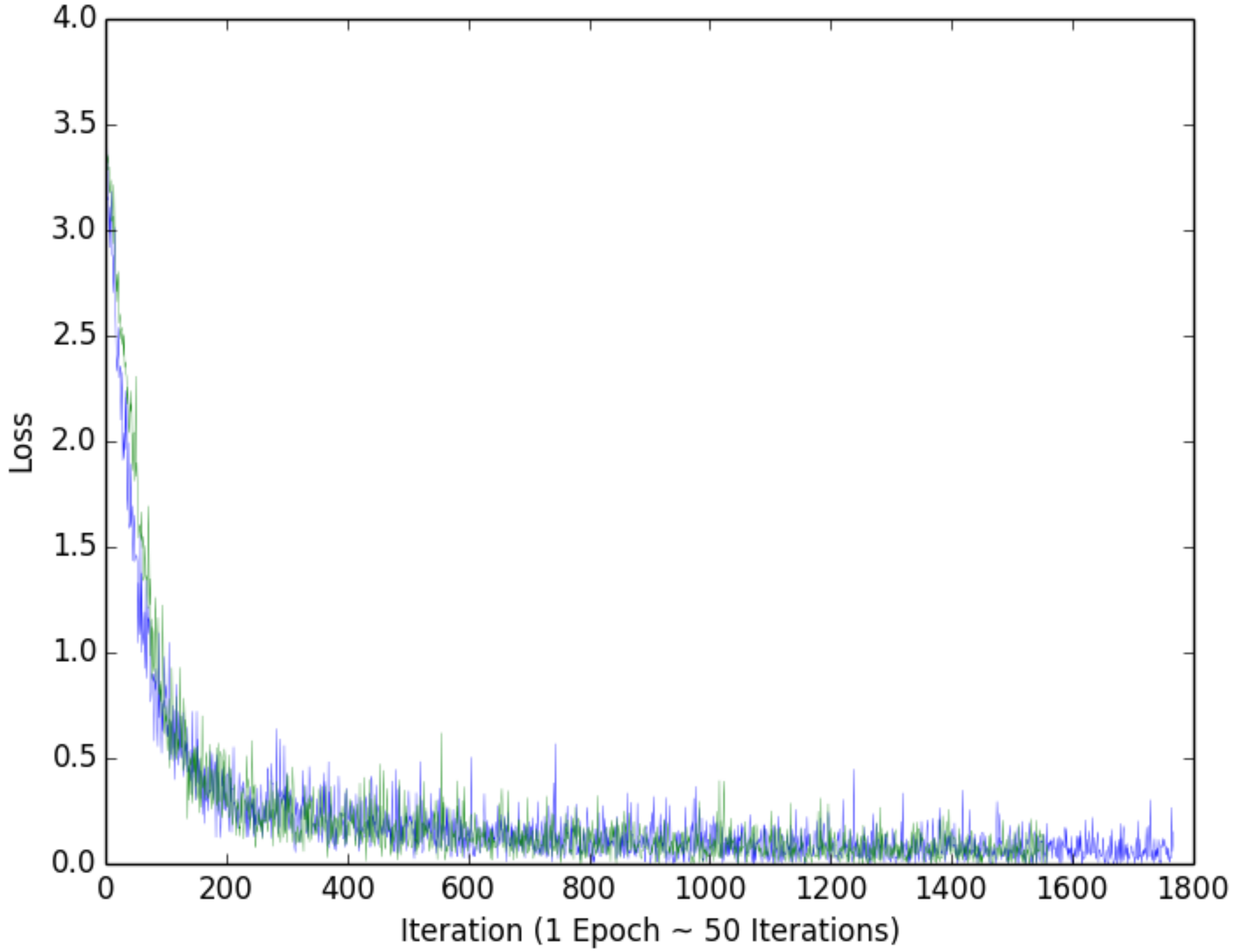
KERNEL SHAPE MODIFICATION

- Kernel shape of the convolutional NN changed to more rectanglar shape
- Augmented from 3x3 to 5x3

- More parameters, slower computation
- Results:

ACCURACY	95.9%
AV. LOSS	0.3423

KERNEL SHAPE MODIFICATION



MIXUP

Beyond Empirical Risk Minimization -ERM-

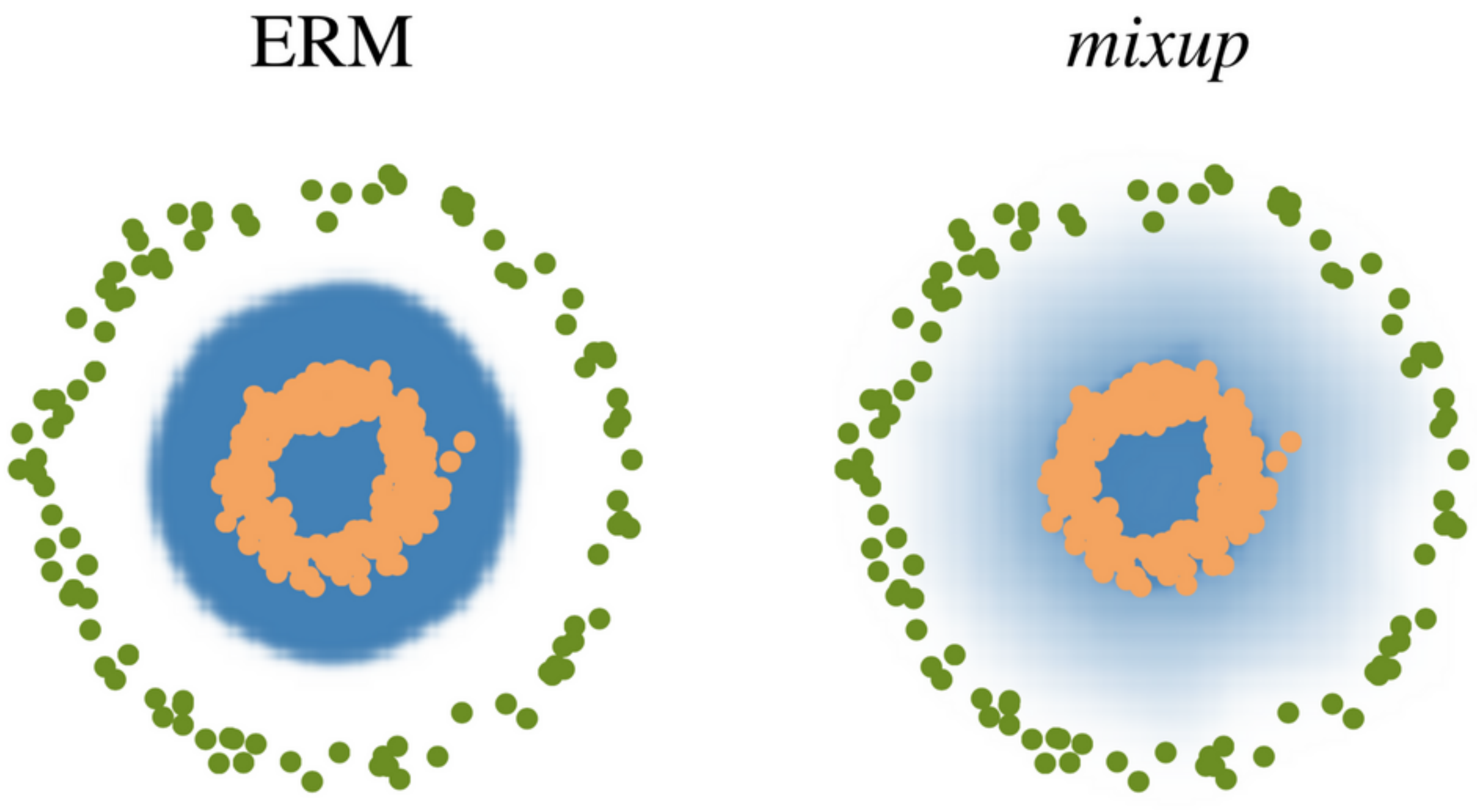
- ERM allows large neural networks to memorize, instead of generalize, the training data
- NN trained with ERM change their predictions when evaluated with data outside the training distribution

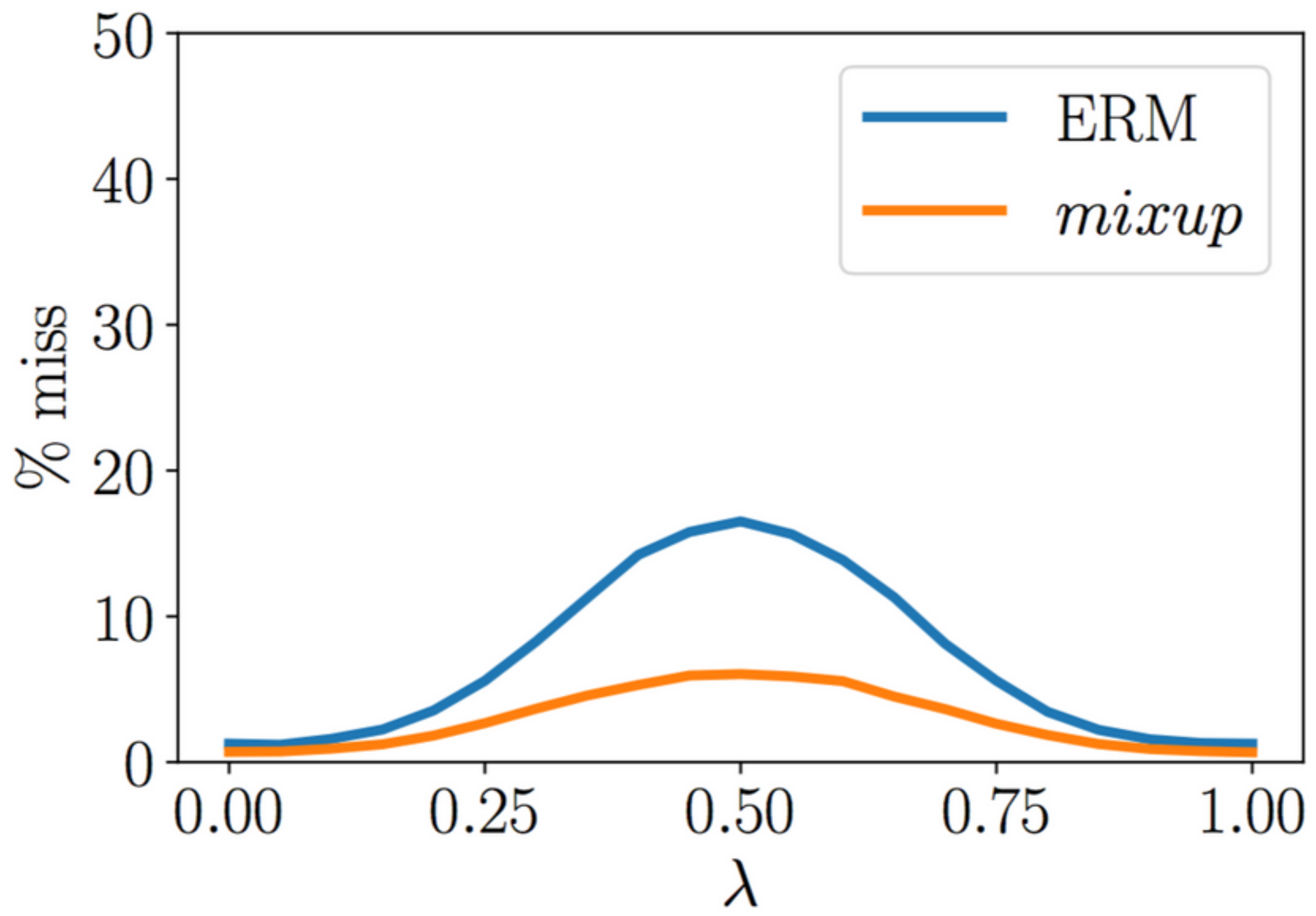
MIXUP: DATA AUGMENTATION

Beyond Empirical Risk Minimization -ERM-

- Formalized as Vicinal Risk Minimization -VRM- principle.
- Increases NN robustness when facing adversaial examples
- Extending training distribution:
    - Linear interpolations of feature vectors (should lead to)
    - Linear interpolations of associated targets

ERM VS MIXUP

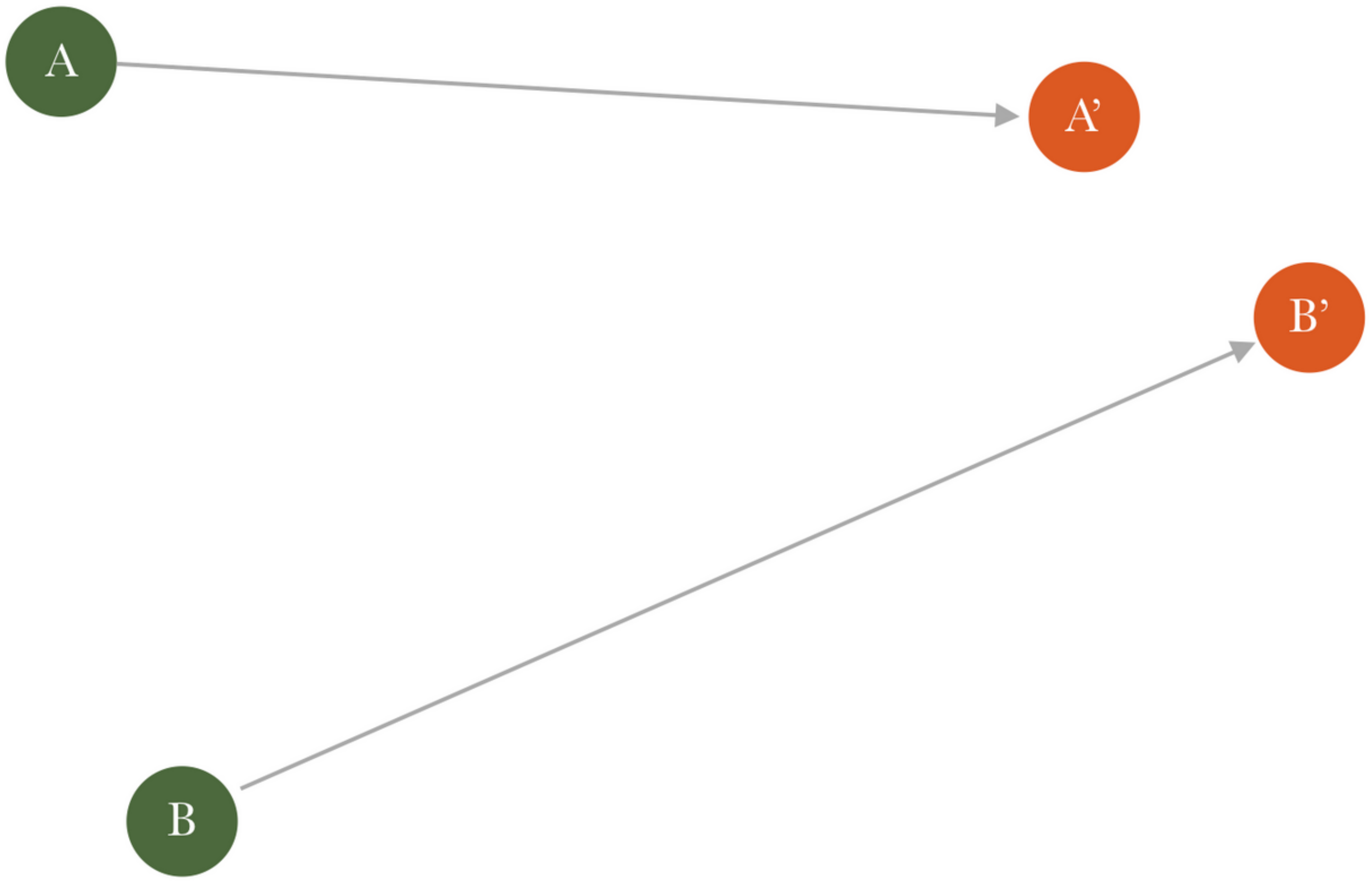




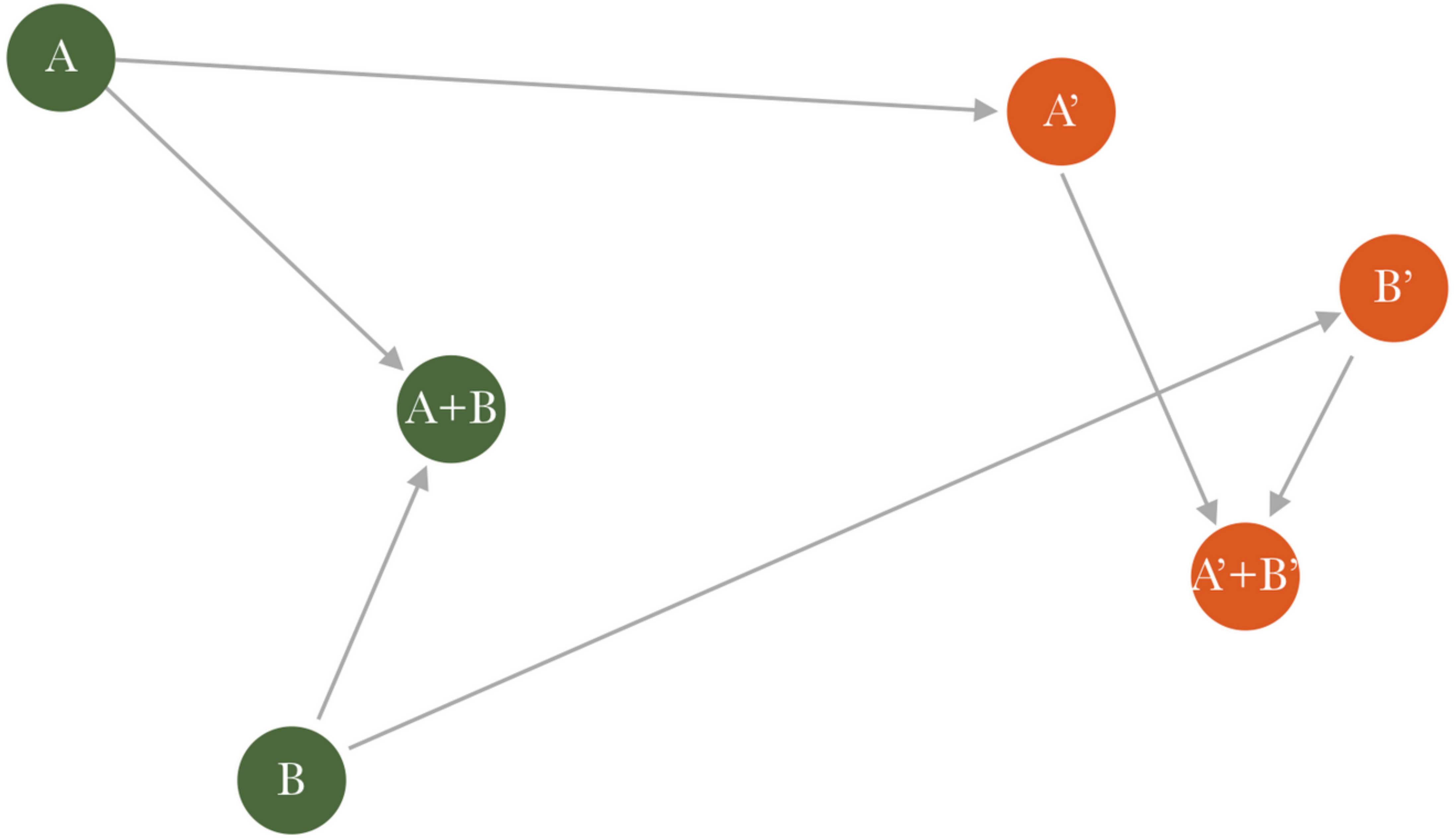
PROPOSED MIXUP DISTRIBUTION

$$\mu(\tilde{x}, \tilde{y}|x_i, y_i) = \frac{1}{n}\sum_{j}^{n} \mathop{\mathbb{E}}_{\lambda}\left[\delta(\tilde{x} = \lambda \cdot x_i + (1-\lambda) \cdot x_j, \tilde{y} = \lambda \cdot y_i + (1-\lambda) \cdot y_j)\right]$$

DATA AUGMENTATION TECHNIQUES: MIXUP

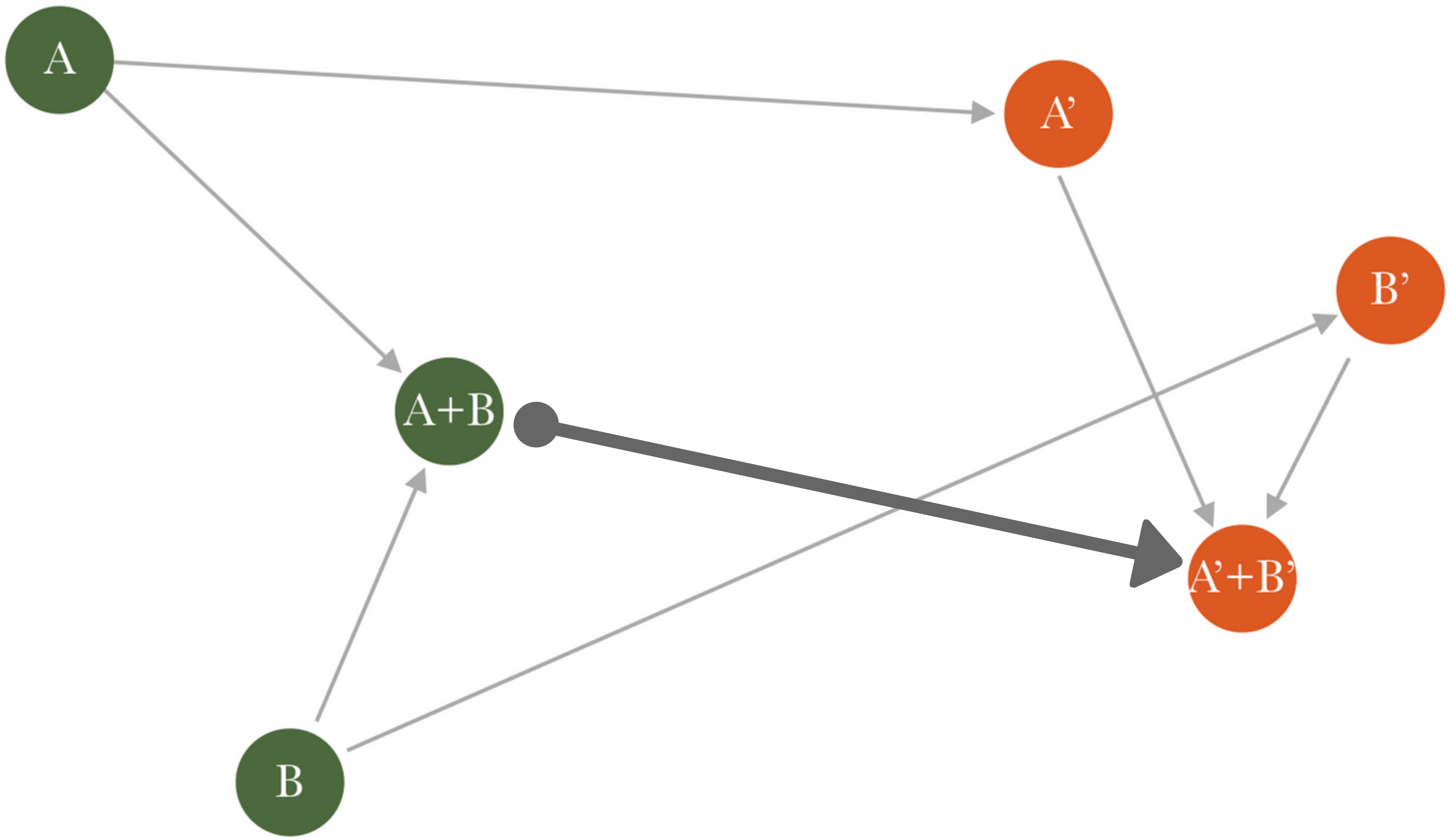


DATA AUGMENTATION TECHNIQUES: MIXUP

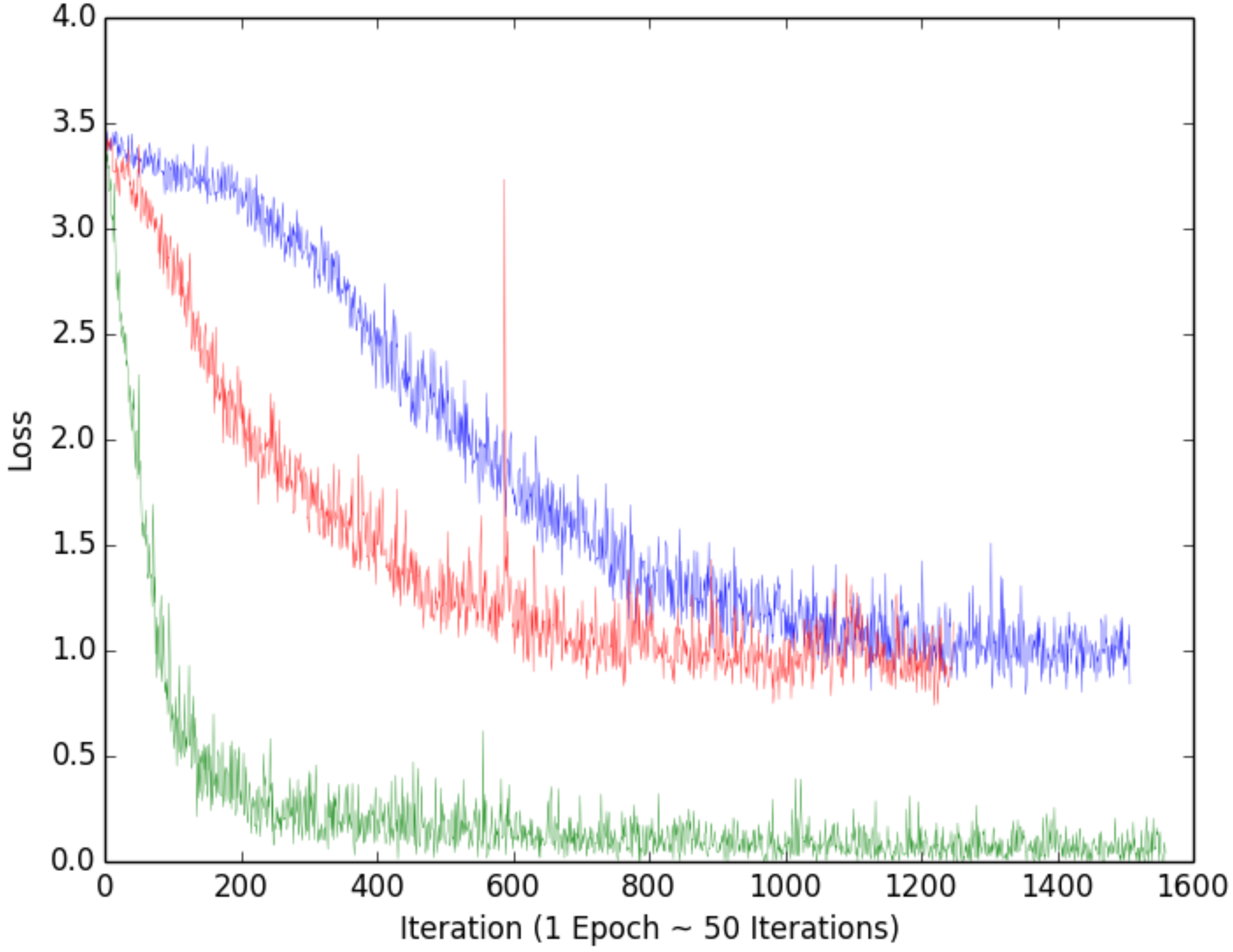


* The sum is representative in the figure. Actually it has been done averaging and stacking.

DATA AUGMENTATION TECHNIQUES: MIXUP



DATA AUGMENTATION TECHNIQUES: MIXUP



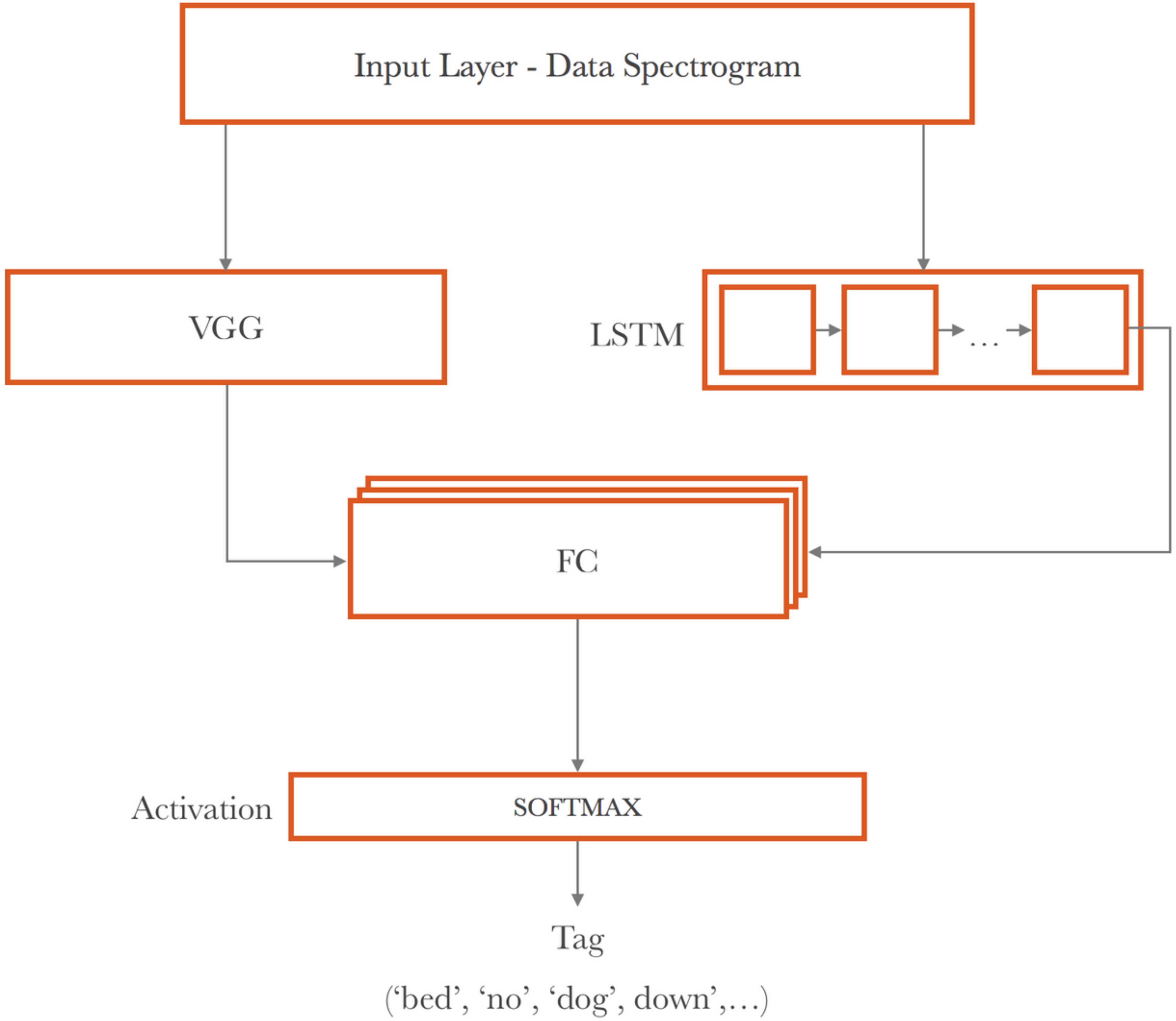
STACKING:

ACCURACY	94.7%
AV. LOSS	0.538

AVERAGING:

ACCURACY	95.9%
AV. LOSS	0.342

VGG PARALLEL TO A LSTM



THANK YOU

REFERENCES

[1] Kaggle. https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/discussion/47715

[2] Gcommands pytorch code. https://github.com/jarfo/gcommands

[3] Speech Commands Dataset. https://research.googleblog.com/2017/08/launching-speech-commands-dataset.html

[4] Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] TensorFlow Speech Recognition Challenge. https://www.kaggle.com/c/tensorflow-speech-recognition-challenge
.