Low-shot Visual Recognition by Shrinking and Hallucinating Features

Bharath Hariharan and Ross Girshick
Facebook AI Research (FAIR)

Presenter – Ashish Kumar

Low shot learning – Related Work

- Generative
  - Hand crafted features
  - Additional attributes
  - Infer using Bayesian Reasoning
- Metric Learning
- Zero shot learning (textual or attribute level description)
- Transfer Learning
- Meta Learning

Low shot learning

- Generative – hallucinate new examples
- Metric Learning – Learning better representations

Learner

Low shot learning – 2 Training Phases, 1 Test

1. Representation learning –

- base categories C_{base},
- A dataset D containing a large number of examples for each category in C_{base}.

The learner uses D to set the parameters of its feature extractor

Low shot learning – 2 Training Phases, 1 Test

2. Low shot learning phase –

- Categories $C_l = C_{base} \cup C_{novel}$
- For each novel category, the learner has access to only n positive examples, where n ∈ {1, 2, 5, 10, 20} (and D for C_{base})

The learner sets the parameters of its multi-class classifier while also optionally modifying the feature extractor.

Low shot learning – 2 Training Phases, 1 Test

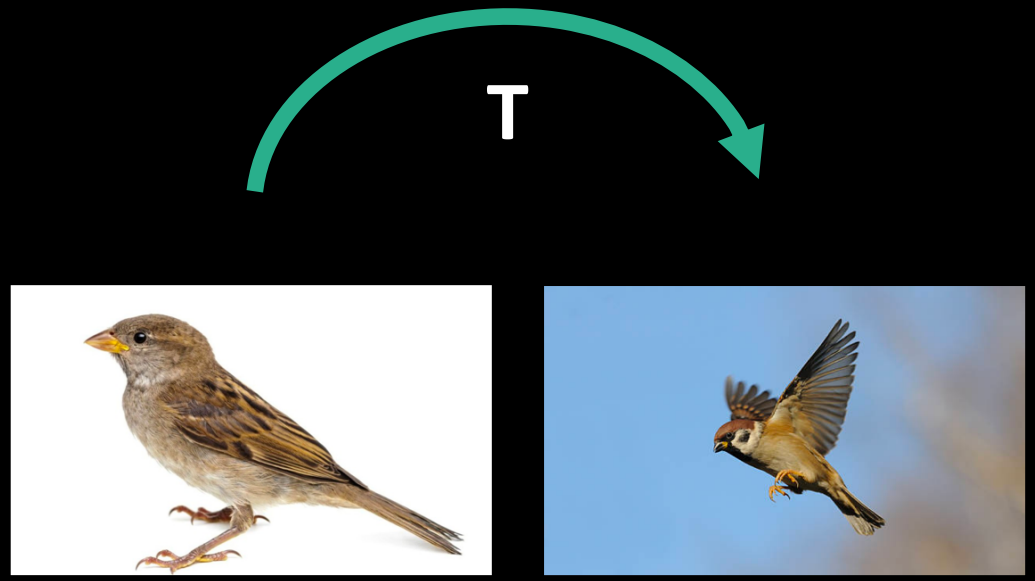
3. Testing phase – the learnt model predicts labels from C_{base} ∪ C_{novel} on a set of previously unseen test images

Low-Shot Learning Through Generation

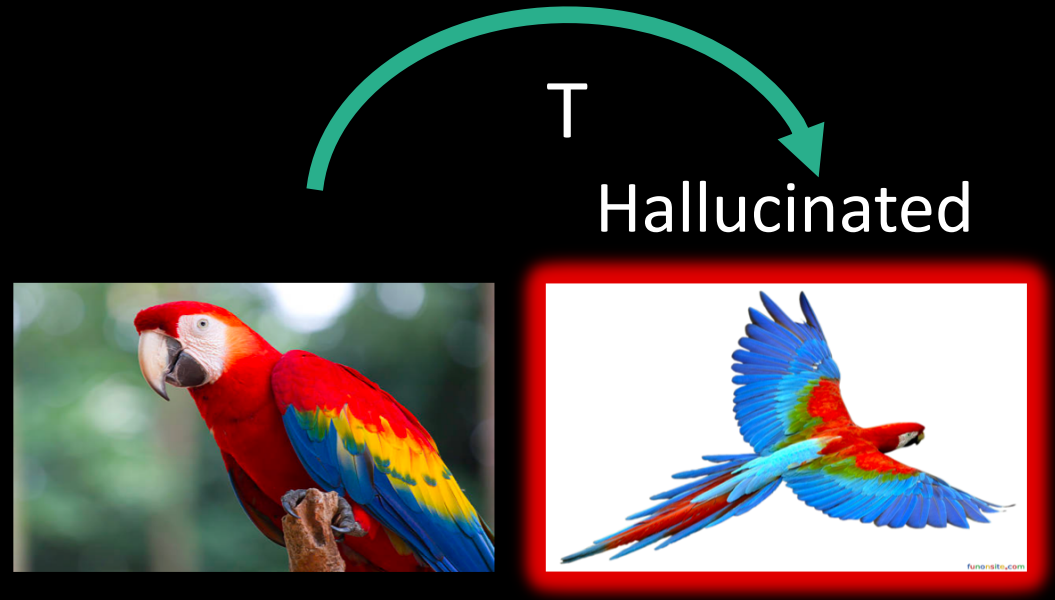
Species 1 from Base Category

Species 2 from Novel Class

Low-Shot Learning Through Generation



Species 1 from Base Category



Species 2 from Novel Class

Low-Shot Learning Through Generation

- More concretely, given x_1, x_2 from one category and z_1 from another, we want to find out

$$x_1 : x_2 :: z_1 : ?$$

Low-Shot Learning Through Generation

- More concretely, given x_1, x_2 from one category and z_1 from another, we want to find out

$$x_1 : x_2 :: z_1 : ?$$

- *Actual generation happens in feature space !*

Generate New Examples

Learning to Generate New Examples

- Cluster the feature vectors in each base category into a fixed number of clusters (100)
- Find $(c_1^a, c_2^a, c_1^b, c_2^b)$ such that:

  cosine similarity between $(c_1^a - c_2^a)$ and $(c_1^b - c_2^b) > 0$

  where, a, b are 2 different categories

- Lets call this set of quadruplets D_G

Learning to Generate New Examples

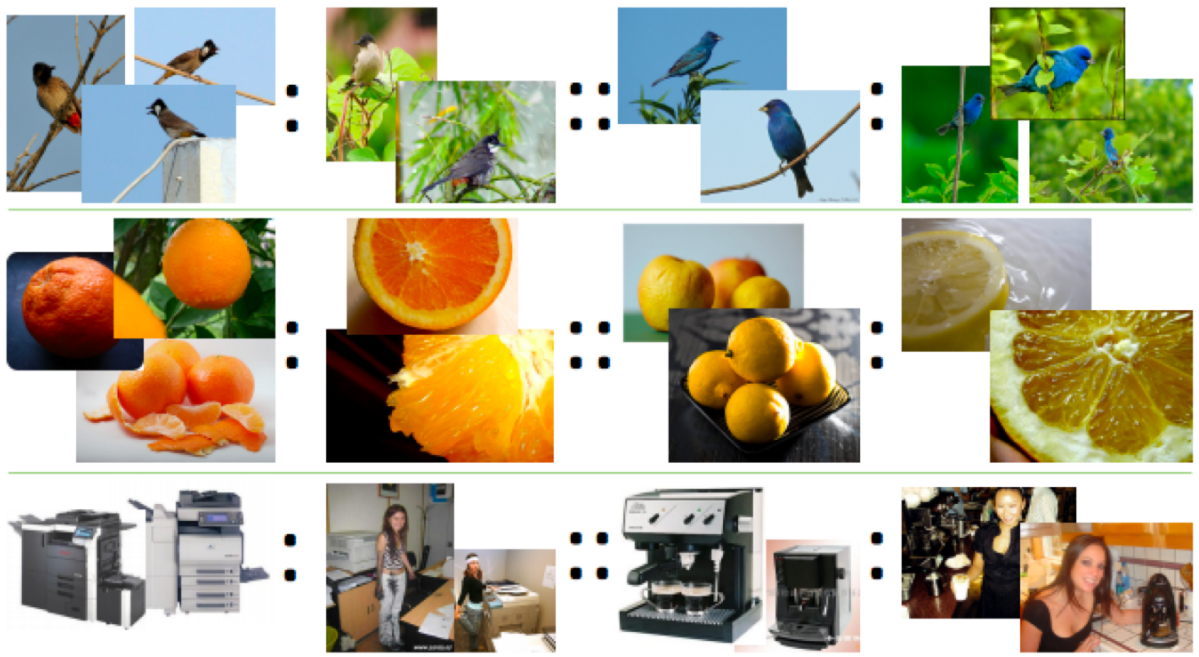


Figure 2: Example mined analogies. Each row shows the four image clusters that form the four elements in the analogy. **Row 1:** birds with a sky backdrop vs birds with greenery in the background. **Row 2:** whole fruits vs cut fruit. **Row 3:** machines (printer, coffee making) in isolation vs the same machine operated by a human.

Learning Procedure for G

- For each quadruplet $(c_1^a, c_2^a, c_1^b, c_2^b)$,

  Feed -- input = (c_1^a, c_1^b, c_2^b) and output = c_2^a.

- Let $\hat{c}_2^a = G(c_1^a, c_2^a, c_1^b, c_2^b)$ be the output of the generator

Learning Procedure for G

- Objective -- $\lambda L_{mse}\left(\widehat{c_2^a}, c_2^a\right) + L_{cls}(W, \hat{c}_2^a, a\ )$

1. $L_{mse}(\hat{c}_2^a, c_2^a) - Mean\ Sq\ Error$
2. $L_{cls}(W, \hat{c}_2^a, a\ )$ – Log Loss of linear classifier W (which is fixed from representation learning phase)

Using G at Test

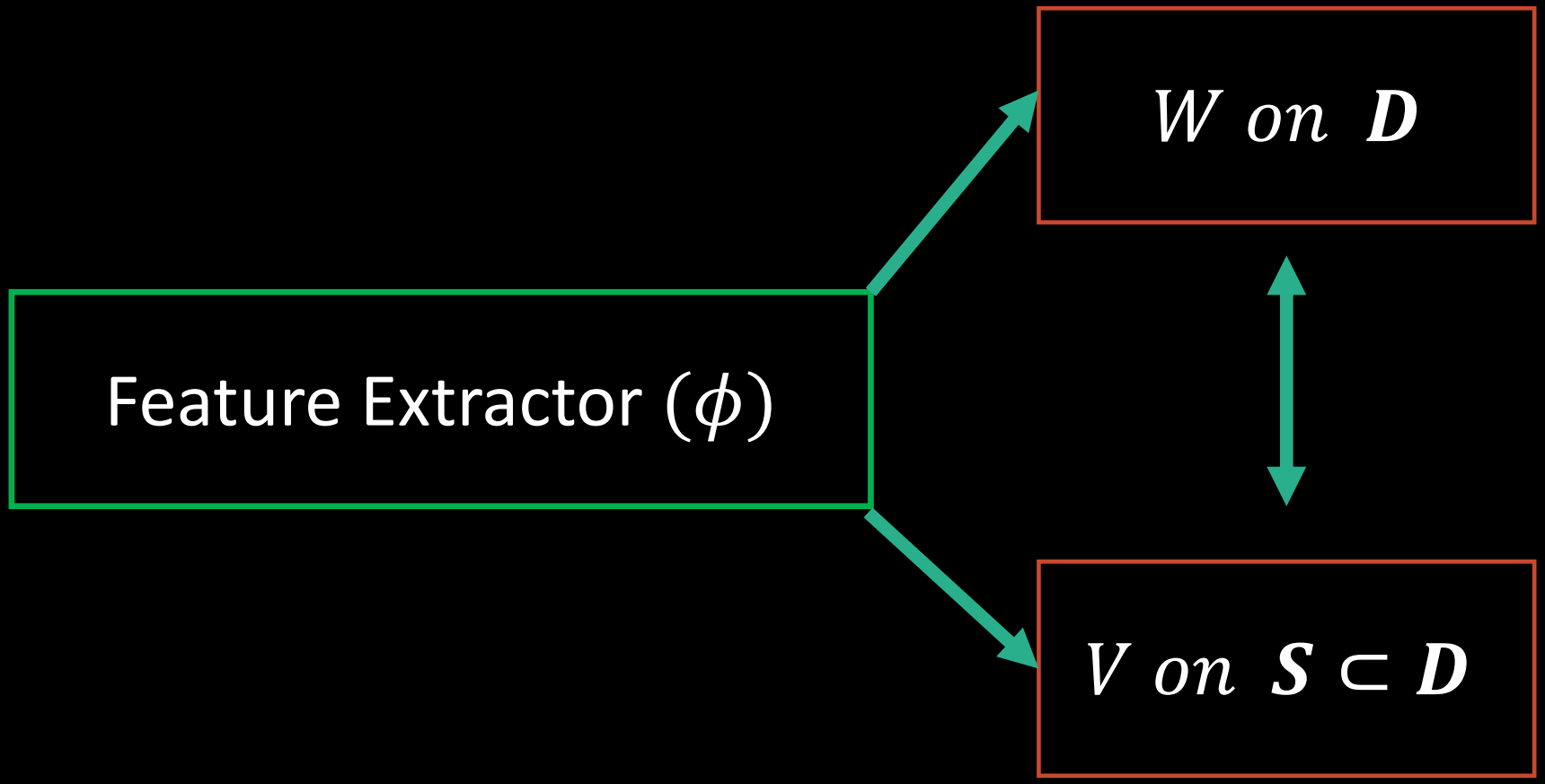
- Given n examples at test, if $n < k$, add $n - k$ examples (k is a hyper-parameter).
- Examples are hallucinated from a random category

Low-Shot Learning via Better Representations

- Let ϕ be the feature extractor, and W be the linear classifier trained on the set of base classes D.

- Objective – Reduce the difference between classifiers trained on D and classifiers trained on $S \subset D$

Low-Shot Learning via Better Representations



Low-Shot Learning via Better Representations

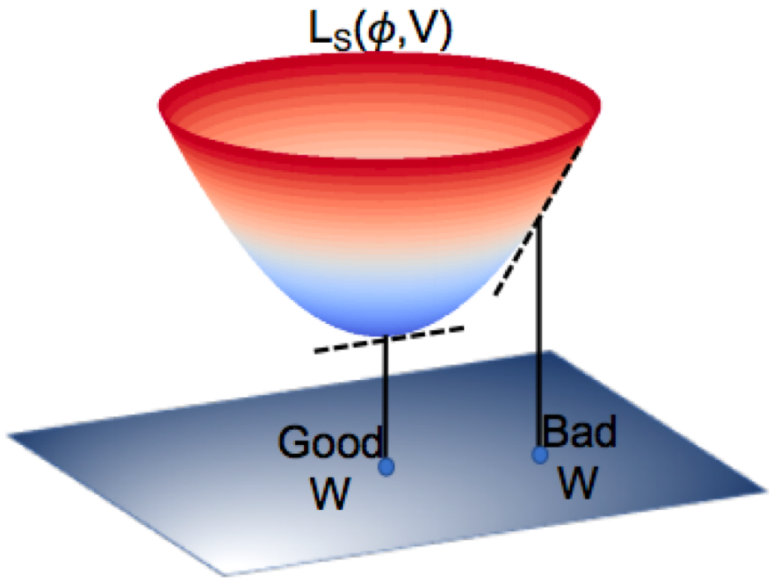


Figure 3: Motivation for the SGM loss. We want to learn a representation ϕ such that the arg min of the small set training objective $L_S(\phi, V)$ matches W, the classifier trained on a large dataset D.

Low-Shot Learning via Better Representations

- Equivalently,
- If we minimize the following to get V,

$$L_S(\phi, V) = \sum_{(x,y)\in S} L_{cls}(V, \phi(x), y)$$

- We want $\nabla_V L_S(\phi, V)|_{V=W} = 0$

Low-Shot Learning via Better Representations

- More generally, the objective function is -

$$\tilde{L}_S(\phi, W) = \|\nabla_V L_S(\phi, V)|_{V=W}\|^2$$

Low-Shot Learning via Better Representations

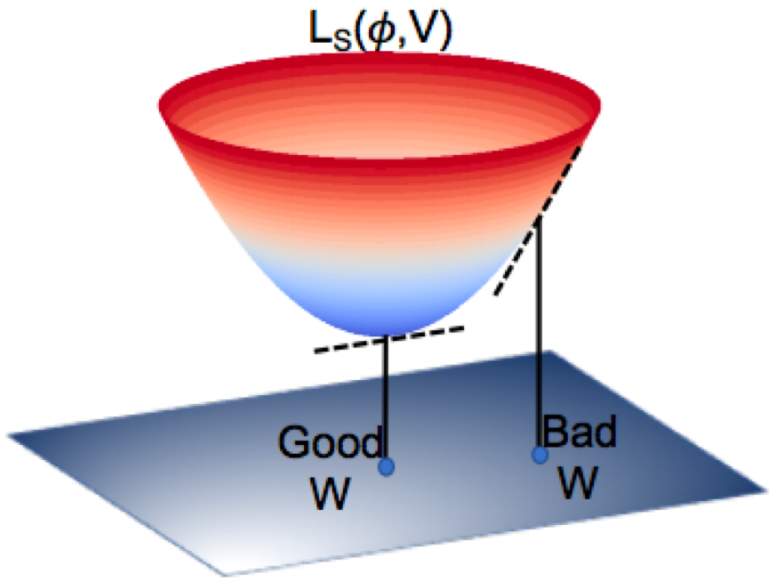


Figure 3: Motivation for the SGM loss. We want to learn a representation ϕ such that the arg min of the small set training objective $L_S(\phi, V)$ matches W, the classifier trained on a large dataset D.

Low-Shot Learning via Better Representations

- Simplified analytically for $|S| = 1$

$$\tilde{L}_S(\phi, W) = \alpha(W, \phi(x), y)\|\phi(x)\|^2$$

Where

- $\alpha(W, \phi(x), y) = \sum_{k=1}^{K}\left(p_k\left(W, \phi(x)\right) - \delta_{yk}\right)^2$
- p_k are soft max probabilities.

Squared Gradient Loss Objective

- We have

$$L^{D}_{SGM}(\phi, W) = \sum_{(x,y)\in D} \alpha(W, \phi(x), y) \ \|\phi(x)\|^2$$

- Final Objective –

$$\min_{W,\phi} L_D(\phi, W) + \lambda L_D^{SGM}(\phi, W)$$

Experimental Setup

- Imagenet1k
- Final numbers on $C^{fin} = C^2_{base}(196\ cl) \cup C^2_{novel}(311\ cl)$
- Used $C^{cv} = C^1_{base}(193\ cl) \cup C^1_{novel}(300\ cl)$ for cross validation

Results

Representation	Lowshot phase	n=1	2	5	10	20
ResNet-10						
Baseline	Classifier	14.1	33.3	56.2	66.2	71.5
Baseline	Generation* + Classifier	29.7	42.2	56.1	64.5	70.0
SGM*	Classifier	23.1	42.4	*61.7*	69.6	*73.8*
SGM*	Generation* + Classifier	*32.8*	46.4	*61.7*	*69.7*	*73.8*
Batch SGM*	Classifier	23.0	42.4	61.9	**69.9**	**74.5**
L1*	Classifier	20.8	40.8	59.8	67.5	71.6
L2*	Classifier	29.1	*47.4*	**62.3**	68.0	70.6
Triplets	Classifier	24.5	41.8	56.0	61.3	64.2
Dropout [20]	Classifier	26.8	43.9	59.6	66.2	69.5
Decov [7]	Classifier	13.0	33.9	59.3	68.9	73.4
Multiverse [30]	Classifier	13.7	30.6	52.5	63.8	71.1
Baseline	Data augmentation	16.0	31.4	52.7	64.4	71.8
Baseline	Model Regression [47]	20.7	39.4	59.6	68.5	73.5
Baseline	Matching Network [46]	**41.3**	**51.3**	*62.1*	67.8	71.8
Baseline-ft	Classifier	12.5	29.5	53.1	64.6	70.4
ResNet-50						
Baseline	Classifier	28.2	51.0	71.0	*78.4*	*82.3*
Baseline	Generation* + Classifier	*44.8*	**59.0**	*71.4*	77.7	*82.3*
SGM*	Classifier	37.8	57.1	**72.8**	**79.1**	**82.6**
SGM*	Generation* + Classifier	**45.1**	*58.8*	*72.7*	**79.1**	**82.6**

Table 1: Top-5 accuracy on only novel classes. Best are bolded and blue; the second best are italicized and red. *Our methods.

Representation	Lowshot phase	n=1	2	5	10	20
ResNet-10						
Baseline	Classifier	43.0	54.3	67.2	72.8	75.9
Baseline	Generation* + Classifier	52.4	59.4	67.5	72.6	76.9
SGM*	Classifier	49.4	60.5	*71.3*	**75.8**	*78.1*
SGM*	Generation* + Classifier	*54.3*	*62.1*	*71.3*	**75.8**	*78.1*
Batch SGM*	Classifier	49.3	60.5	71.4	*75.8*	**78.5**
L1*	Classifier	47.1	58.5	69.2	73.7	76.1
L2*	Classifier	52.7	**63.0**	**71.5**	74.8	76.4
Triplets	Classifier	47.6	57.1	65.2	68.4	70.2
Dropout [20]	Classifier	50.1	59.7	68.8	72.7	74.7
Decov [7]	Classifier	43.3	55.7	70.1	75.4	77.9
Multiverse [30]	Classifier	44.1	54.2	67.0	73.2	76.9
Baseline	Data Augmentation	44.9	54.0	66.4	73.0	77.2
Baseline	Model Regression [47]	46.4	56.7	66.8	70.4	72.0
Baseline	Matching Network [46]	**55.0**	61.5	69.3	73.4	76.2
Baseline-ft	Classifier	41.7	51.7	65.0	71.2	74.5
ResNet-50						
Baseline	Classifier	54.1	67.7	*79.1*	*83.2*	**85.4**
Baseline	Generation* + Classifier	*63.1*	**71.5**	78.8	82.6	**85.4**
SGM*	Classifier	60.0	*71.3*	**80.0**	**83.3**	*85.2*
SGM*	Generation* + Classifier	**63.6**	**71.5**	**80.0**	**83.3**	*85.2*

Table 2: Top-5 accuracy on base and novel classes. Best are bo and blue; the second best are italicized and red. *Our method

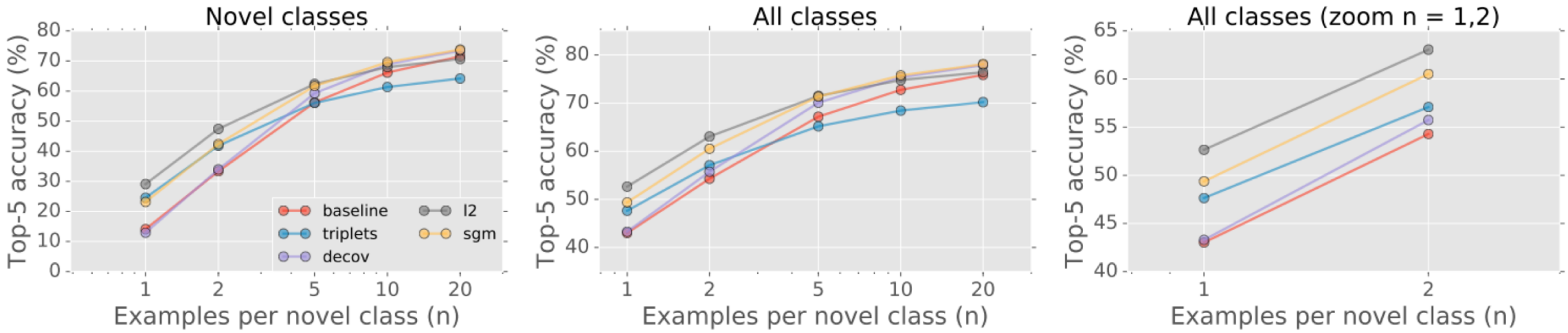


Figure 4: **Representation learning comparison.** Top-5 accuracy on ImageNet1k val. Top-performing feature regularization methods reduce the training samples needed to match the baseline accuracy by 2x. Note the different Y-axis scales.

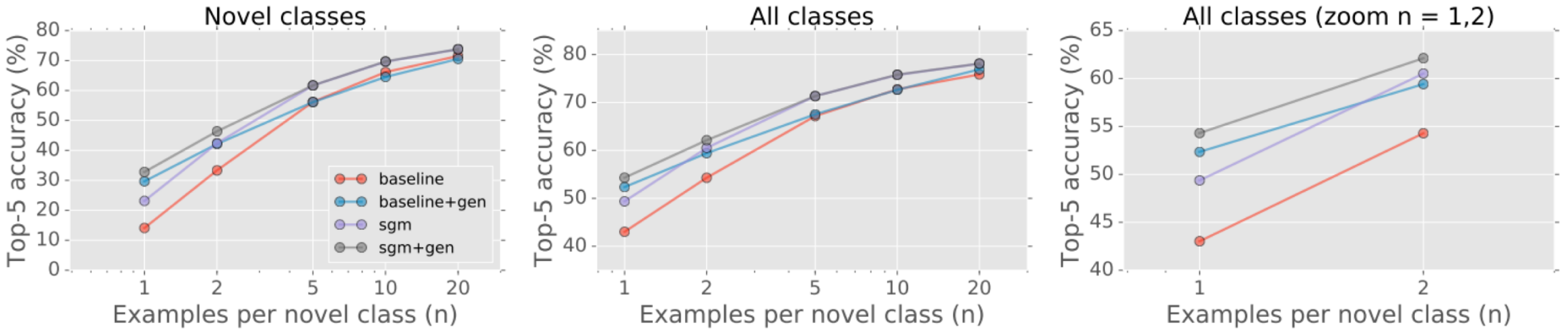


Figure 5: **Comparisons with and without example generation.** Top-5 accuracy on ImageNet1k val. Note the different Y-axis scales.

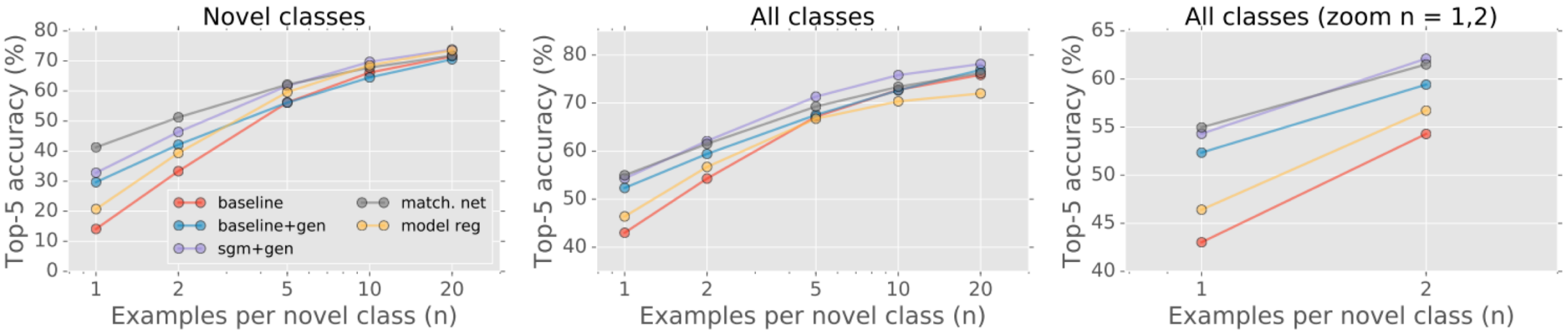


Figure 6: **Comparison to recently proposed methods.** Top-5 accuracy on ImageNet1k val. Note the different Y-axis scales.

Thank You