FixMatch em CIFAR-10: Uma abordagem experimental

Tomás Lira, Gabriel Abad, Antônio Brych

FGV EMAp

13 de novembro de 2025

Visão Geral do Modelo

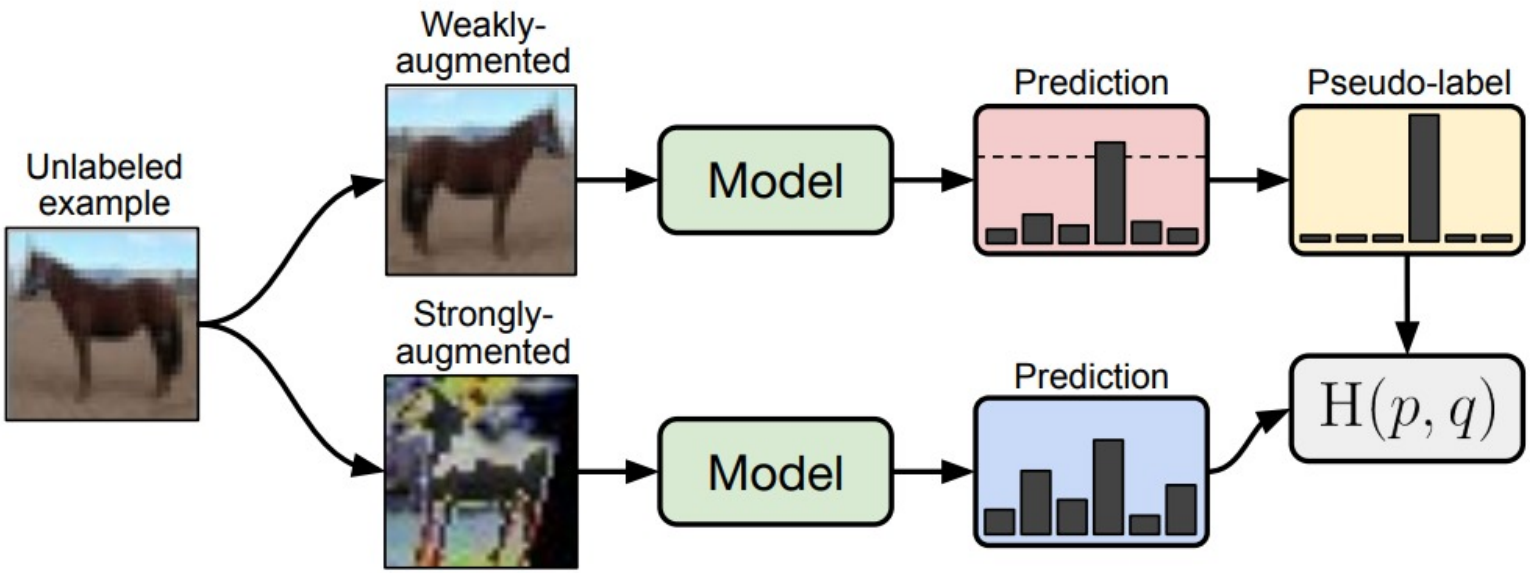


Figura: Fonte: Sohn et al., 2025. FixMatch: Simplifying semi-supervised learning with consistency and confidence.

Algoritmo do FixMatch

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ {*Cross-entropy loss for labeled data*}

for $b = 1$ **to** μB **do**

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ {*Compute prediction after applying weak data augmentation of* u_b}

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), p_{\mathrm{m}}(y \mid \mathcal{A}(u_b))$ {*Cross-entropy loss with pseudo-label and confidence for unlabeled data*}

return $\ell_s + \lambda_u \ell_u$

Figura: Descrição do algoritmo FixMatch (Sohn et al., 2020).

FixMatch

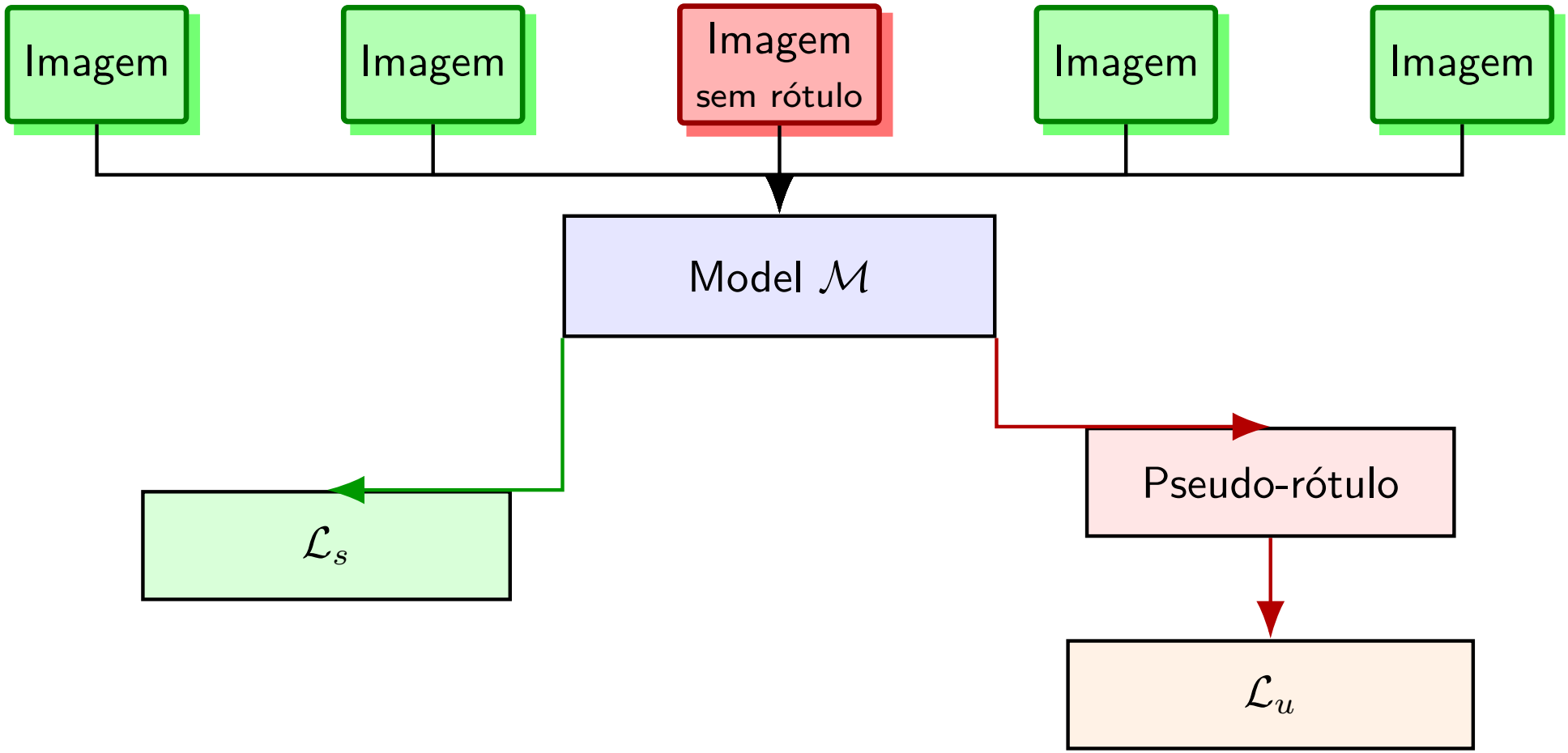
Intuição: Um bom modelo deve ser consistente, isto é, manter a mesma predição para uma mesma imagem sob augmentação forte e fraca.

O FixMatch se baseia nesse princípio para poder utilizar dados não rotulados em seu treinamento.

Questionamento: Por que calcular "pseudo-labels"em dados fracamente augmentados?

Fluxograma (FixMatch): Exemplo em um 'batch'

4 verdes (rotuladas) e 1 vermelha (sem rótulo)



$\mathcal{L}_s$: Perda supervisionada.

$\mathcal{L}_u$: Perda não supervisionada.

FixMatch

$$p_m(y|\mathbf{x}) = \text{softmax}(f_\theta(\mathbf{x}))$$

$$\hat{q}_b = \text{argmax}(p_m(y|\text{AugmentaçãoFraca}(\mathbf{x}_b)))$$

$$m_b = \max\left(p_m(y|\text{AugmentaçãoFraca}(\mathbf{x}_b))\right)$$

$$\mathcal{L}_s = \frac{1}{B}\sum^{B} H(y_b, p_m(y|\mathbf{x}_b))$$

Imposição de consistência

$$\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbf{1}(m_b \geq \tau)\, H(\hat{q}_b, p_m(y|\text{AugmentaçãoForte}(\mathbf{x}_b)))$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_s + \lambda_u \mathcal{L}_u$$

Estrutura Geral da Implementação

- A implementação foi totalmente guiada pelo paper original.
- Organização do código:
  - `make_cifar10_datasets()` – divide o CIFAR-10 em subconjuntos rotulados e não rotulados.
  - `LabeledWrapper` e `UnlabeledWrapper` – classes personalizadas que aplicam as transformações *weak* e *strong*.
  - `get_model()` – cria a **ResNet-18** ajustando a camada final para 10 classes.
  - `ModelEMA` – mantém a média exponencial dos pesos para avaliação estável.

Função de Treino `train_fixmatch`

- Cada **epoch** itera sobre batches rotulados e não rotulados simultaneamente.
- Estrutura principal:
  1. **Forward supervisionado:** calcula a perda de entropia cruzada em (x_l, y_l).
  2. **Forward não supervisionado:** gera pseudo-rótulos a partir de (x_{ul}^{weak}) e aplica a perda nas versões (x_{ul}^{strong}).
  3. Filtra apenas as amostras com confiança $\geq \tau = 0.95$.
  4. Soma ponderada: $\mathcal{L} = \mathcal{L}_s + \lambda_u \mathcal{L}_u$ com $\lambda_u = 1.0$.
- Cada passo atualiza tanto o modelo principal quanto o modelo EMA.

Detalhes das Aumentações

- **Aumentação Fraca (*weak_transform*)**
  - `RandomHorizontalFlip()` – espelhamento aleatório.
  - `RandomCrop(32, padding=4)` – deslocamento até 12,5%.
  - Normalização CIFAR-10: médias e desvios padrão padrão.
- **Aumentação Forte (*strong_transform*)**
  - `RandAugment(num_ops=2, magnitude=9)` – aumenta variação sem reduzir qualidade.
  - `Cutout(length=8)` – remove regiões aleatórias da imagem (regularização espacial).

.

EMA

- A classe **`ModelEMA`** mantém uma cópia dos pesos atualizada a cada passo:

$$\theta_{\text{EMA}} \leftarrow \alpha\,\theta_{\text{EMA}} + (1-\alpha)\,\theta \quad \text{com } \alpha = 0.999$$

- O modelo EMA é usado para avaliação final, garantindo previsões mais estáveis.
- EMA foi utilizado pois é uma recomendação, feita no paper original, para reportar os resultados.

Decisões e Hiperparâmetros

- **Modelo:** ResNet-18 pré-definida do `torchvision`.
- **Augmentation:** RandAugment + Cutout (forte); Flip + Crop (fraco).
- **Limiar de confiança:** $\tau = 0.95$
- **Otimizador:** SGD + Nesterov momentum (0.9)
- **Scheduler:** Decaimento cossenoide da taxa de aprendizado.
- **EMA:** Decaimento 0.999, usado apenas na avaliação.
- **Batch:** $B = 64$, $\mu = 7$

Framework Experimental

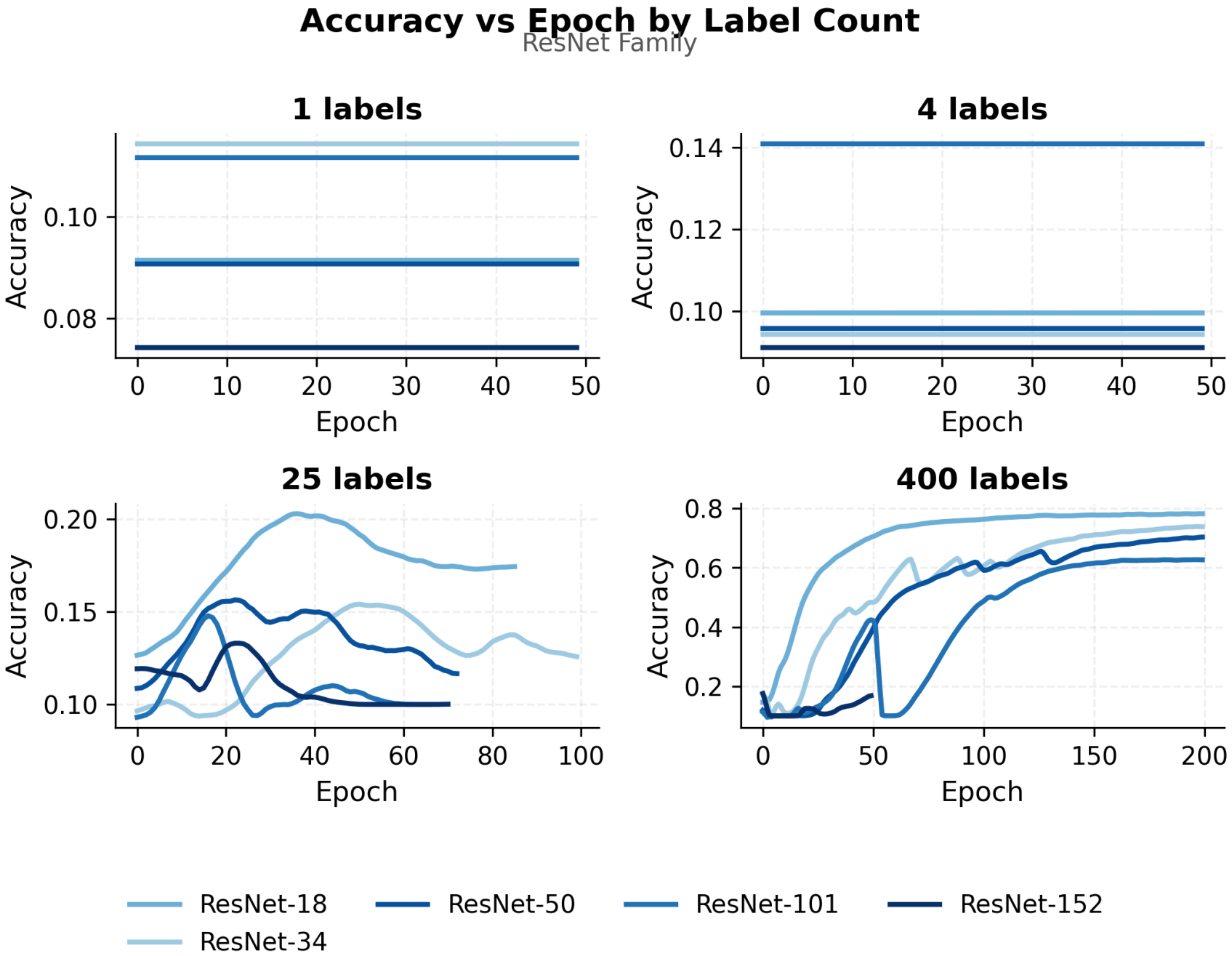
Experimentos com 8 modelos distintos!

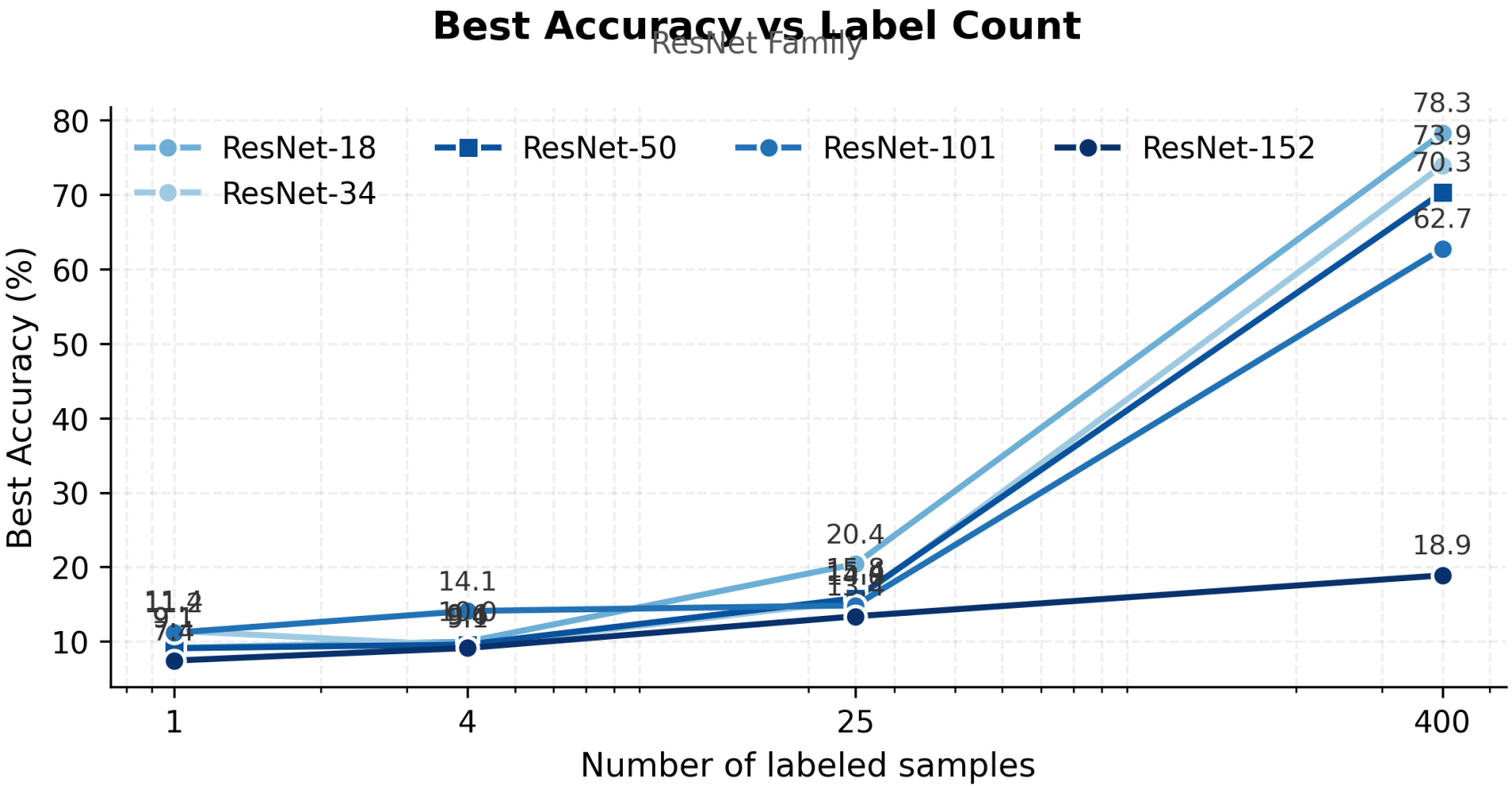
- **Família ResNet**: ResNet-18; ResNet-32; ResNet-34; Resnet-50; ResNet-101; ResNet-152;
- **Outros Modelos**: DenseNet-121; MobileNetV3

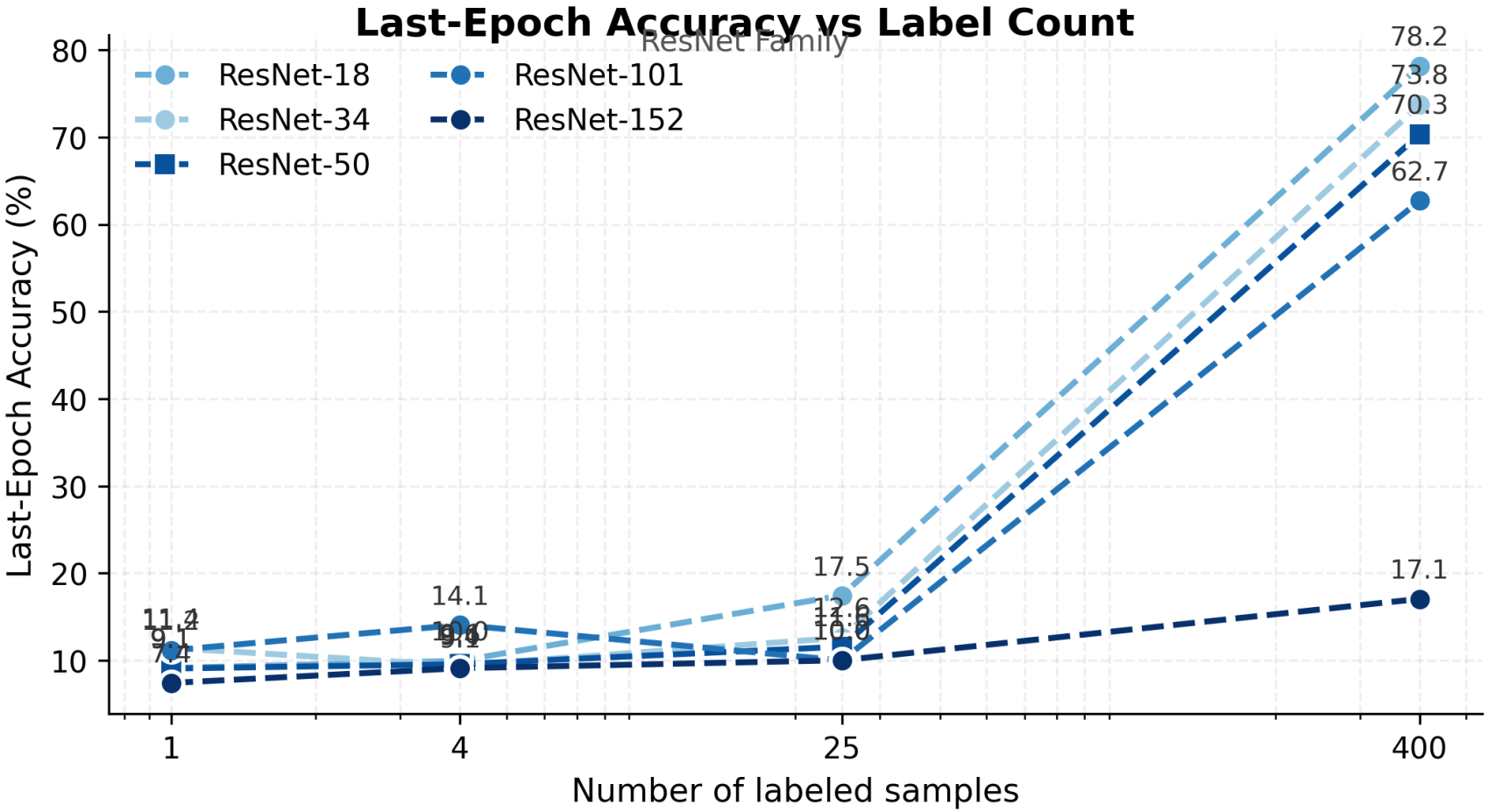
Testes realizados:

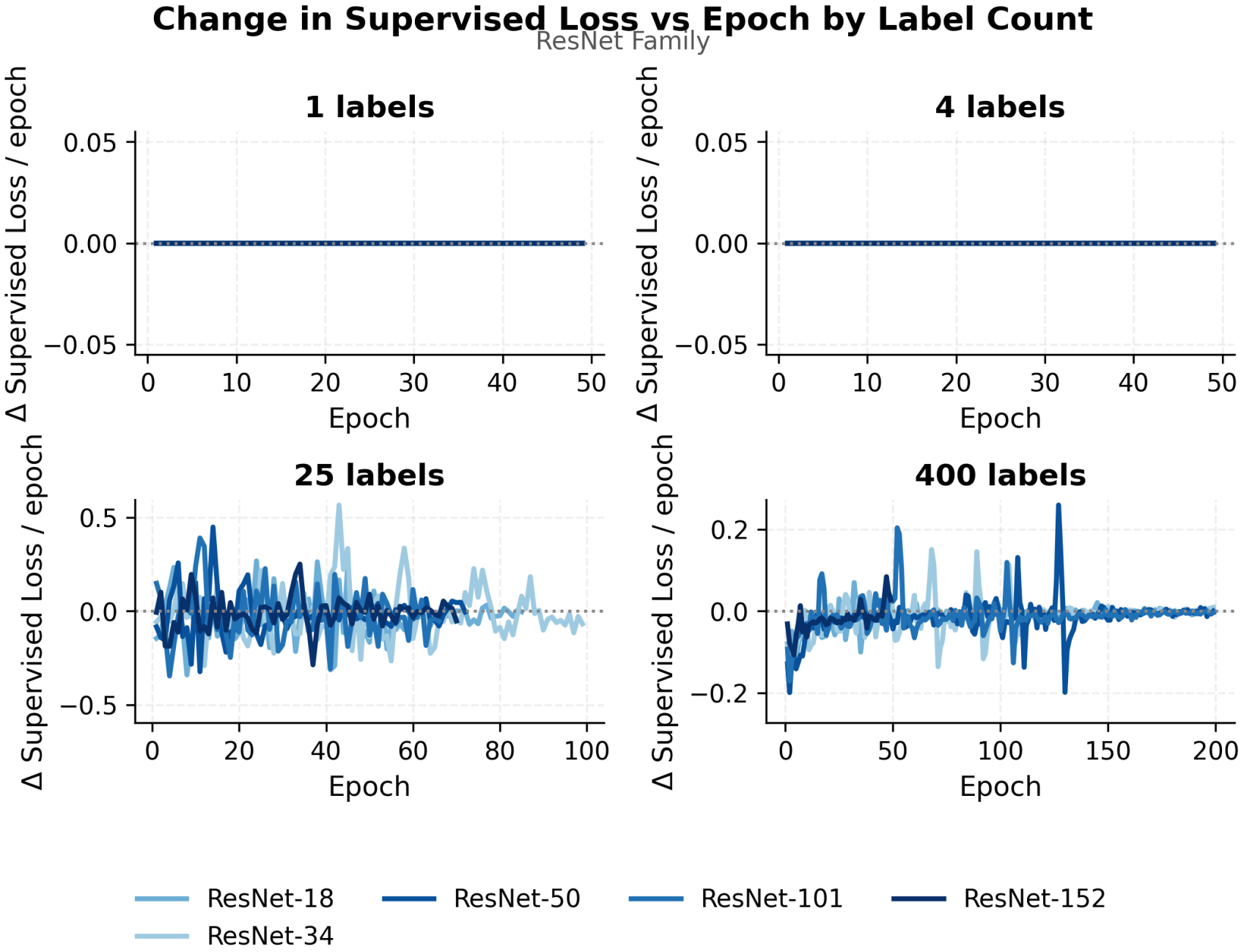
- Testes de Convergência
- Evoluções das curvas de acurácia
- Comportamento da 'Mask-Rate'

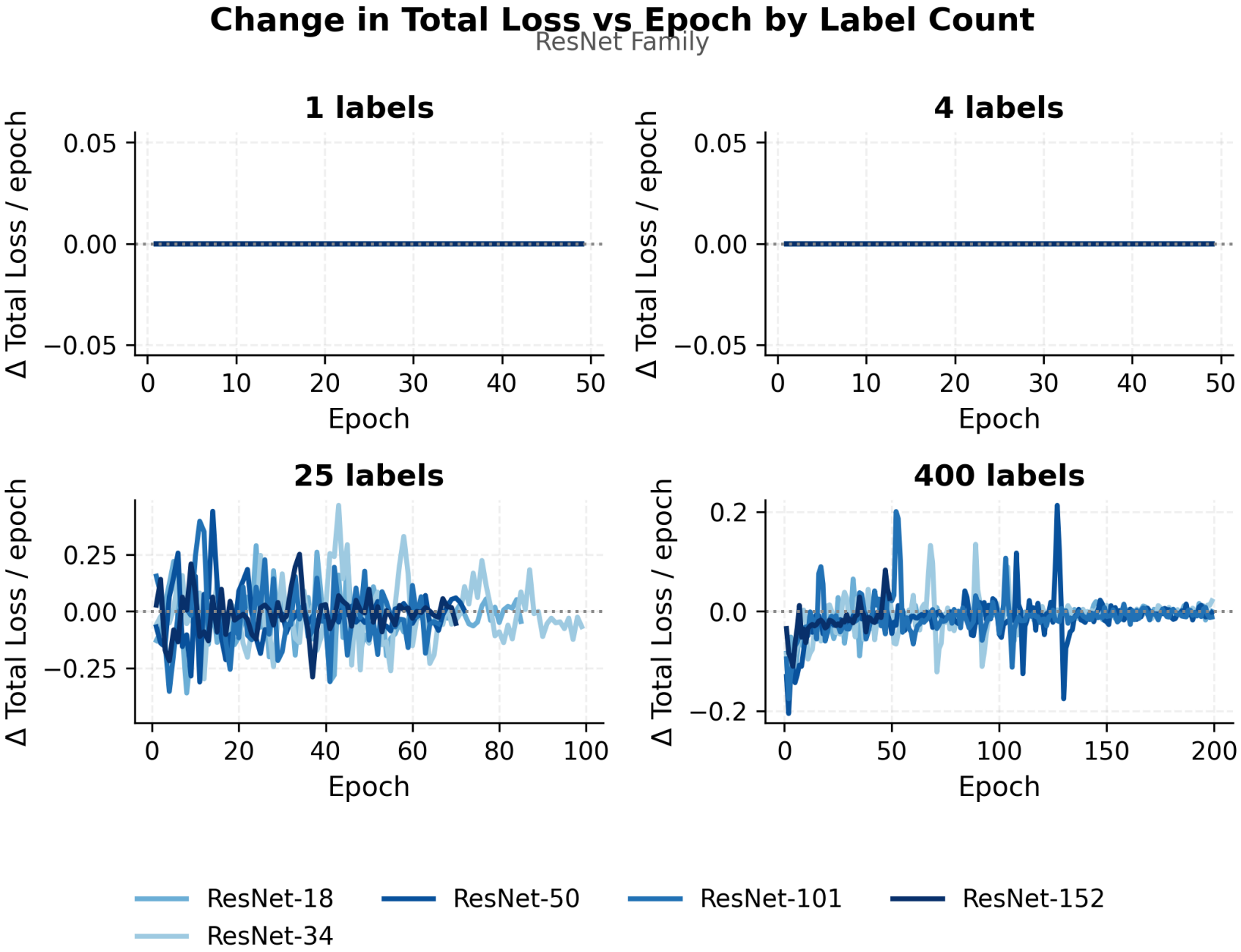
ResNets

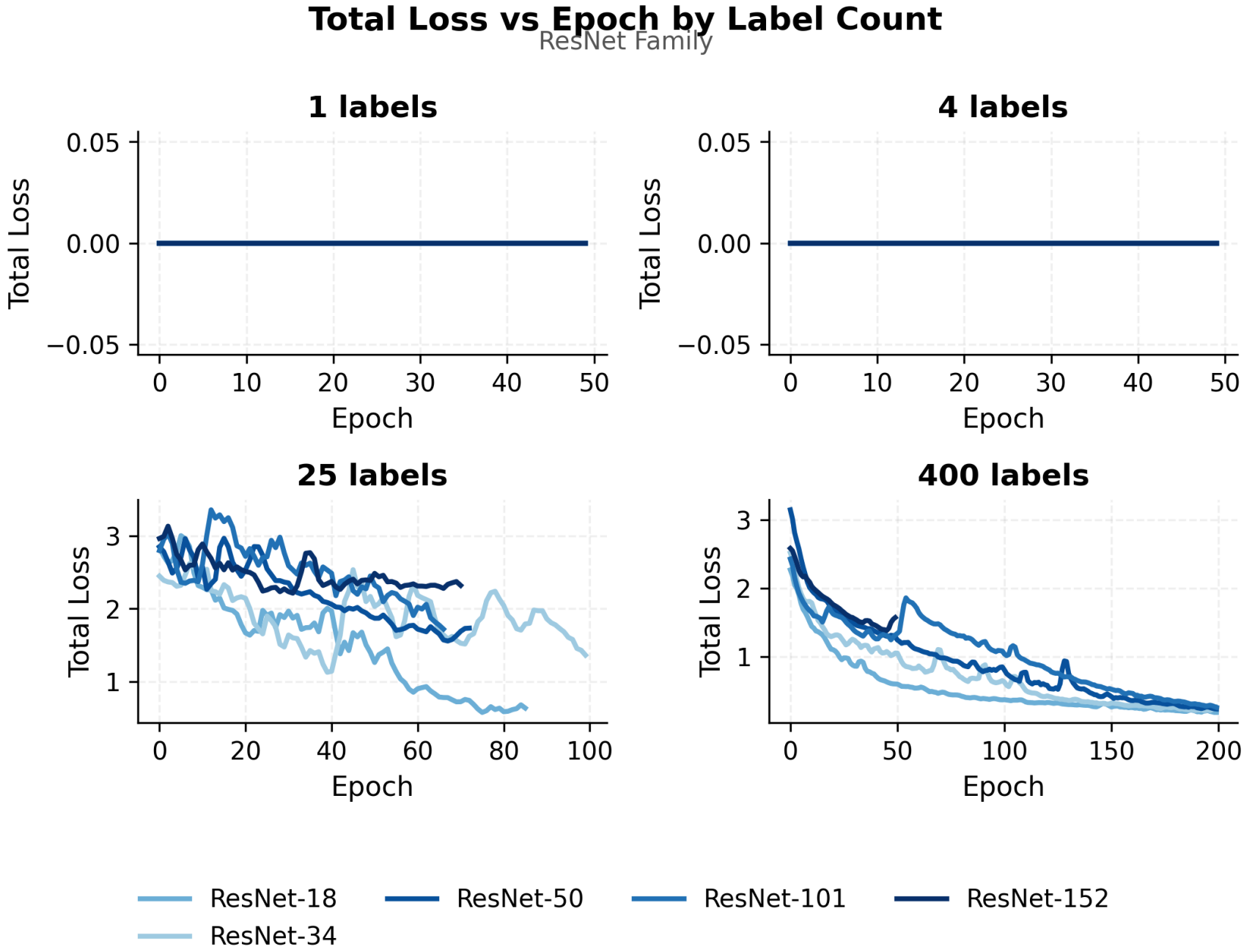

Accuracy vs Epoch by Label Count
ResNet Family
1 labels
4 labels
25 labels
400 labels
Accuracy
Epoch
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152

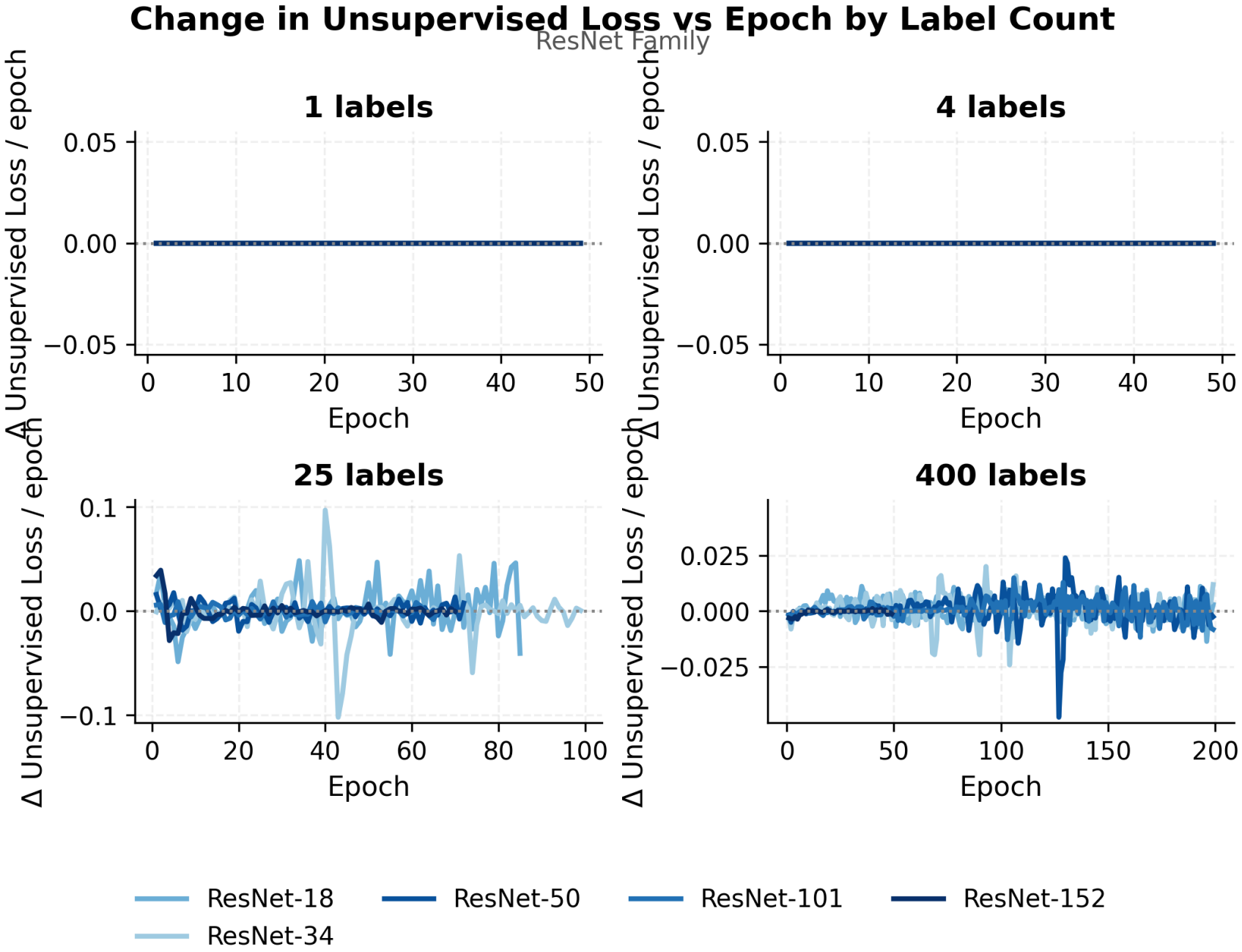

Best Accuracy vs Label Count
ResNet Family
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Best Accuracy (%)
Number of labeled samples
78.3
73.9
70.3
62.7
20.4
18.9
14.1


Last-Epoch Accuracy vs Label Count
ResNet Family
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Last-Epoch Accuracy (%)
Number of labeled samples
80
70
60
50
40
30
20
10
1
4
25
400
78.2
73.8
70.3
62.7
17.5
17.1
14.1
11.4

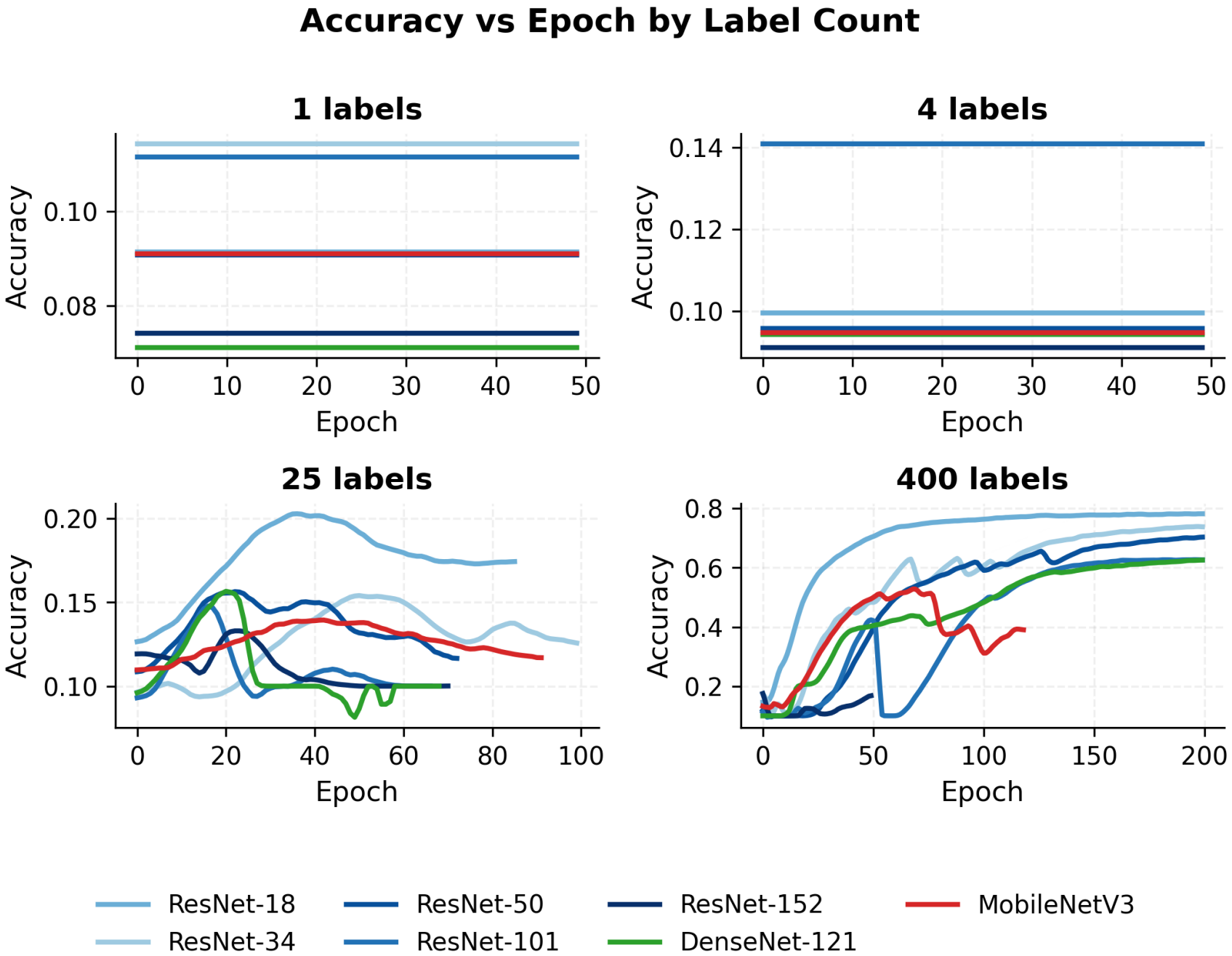

Change in Supervised Loss vs Epoch by Label Count
ResNet Family
1 labels
4 labels
25 labels
400 labels
Δ Supervised Loss / epoch
Epoch
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152

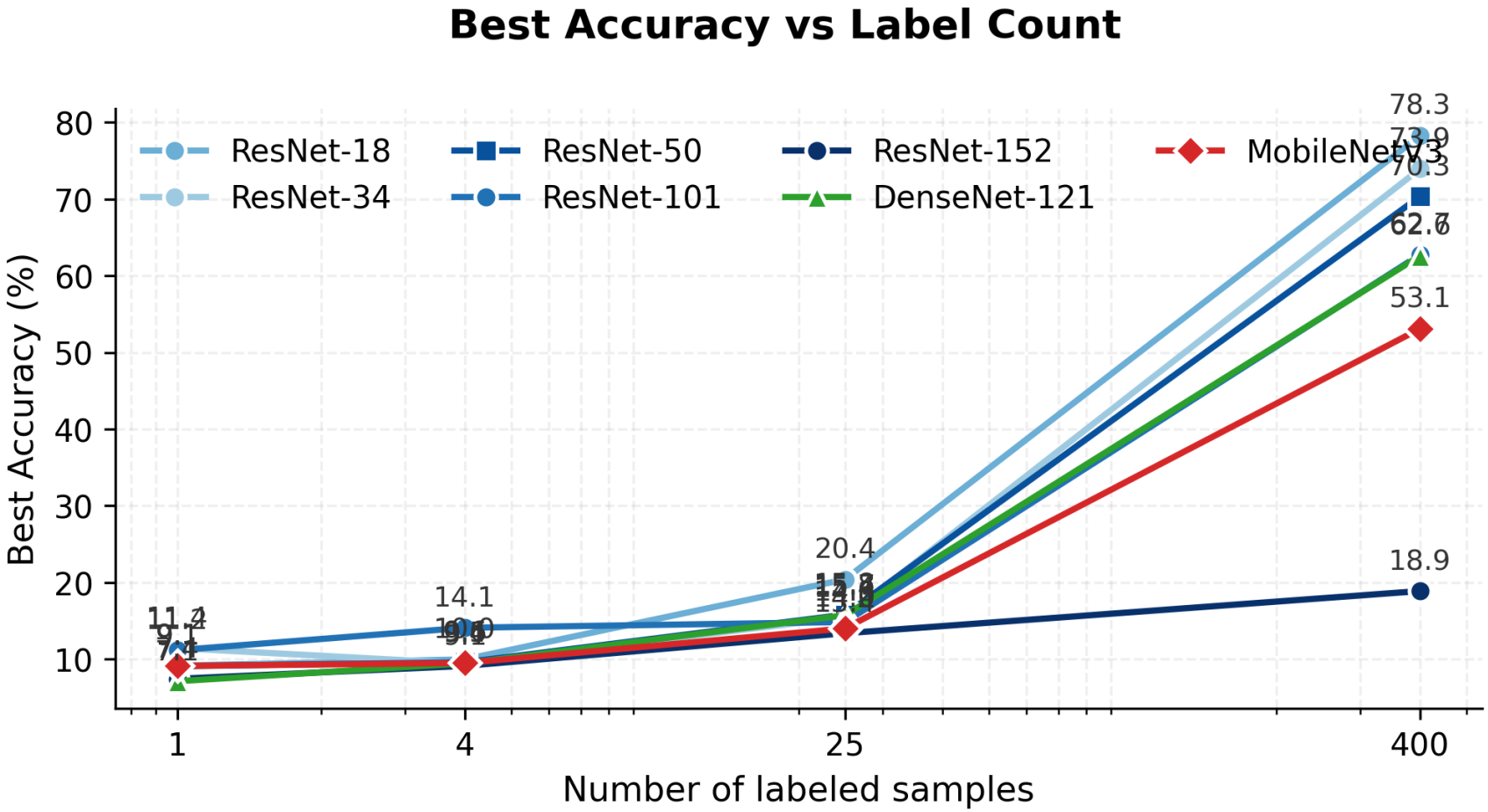

Change in Total Loss vs Epoch by Label Count
ResNet Family
1 labels
4 labels
25 labels
400 labels
Δ Total Loss / epoch
Epoch
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152

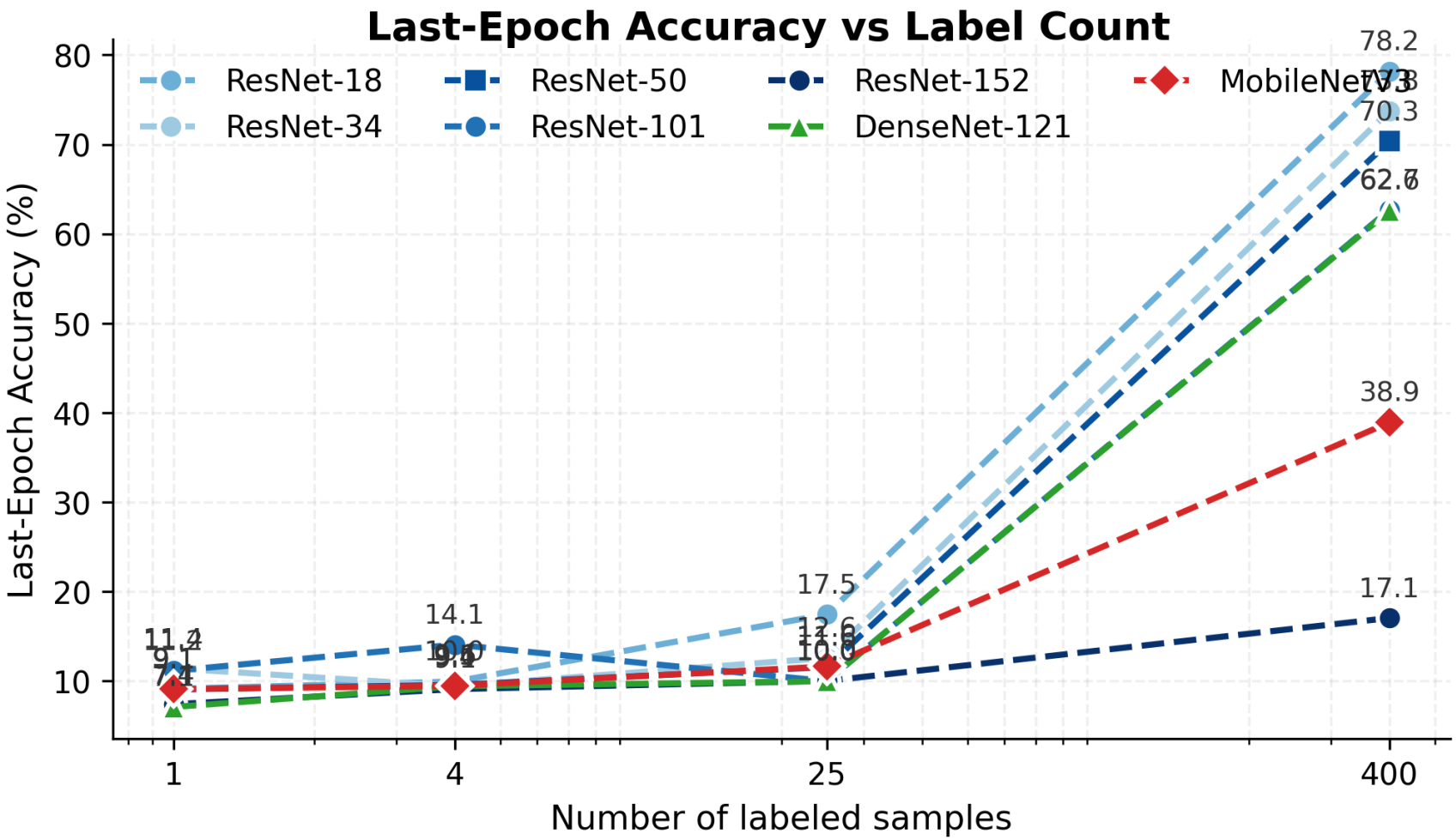

Total Loss vs Epoch by Label Count
ResNet Family
1 labels
4 labels
25 labels
400 labels
Total Loss
Epoch
0.05
0.00
−0.05
0
10
20
30
40
50
1
2
3
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152

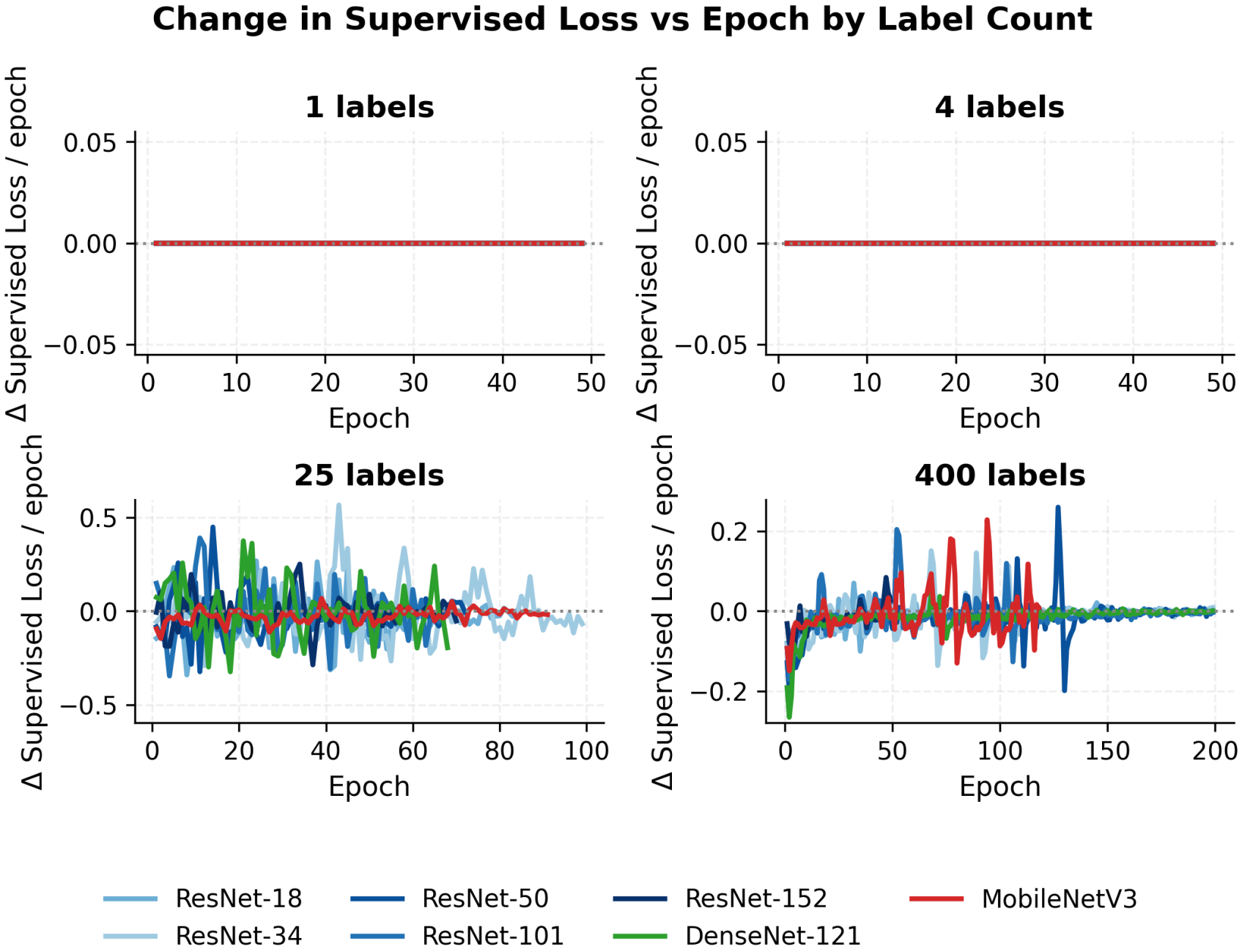

Change in Unsupervised Loss vs Epoch by Label Count
ResNet Family
1 labels
4 labels
25 labels
400 labels
Δ Unsupervised Loss / epoch
Epoch
0.05
0.00
−0.05
0.1
0.0
−0.1
0.025
0.000
−0.025
0
10
20
30
40
50
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152

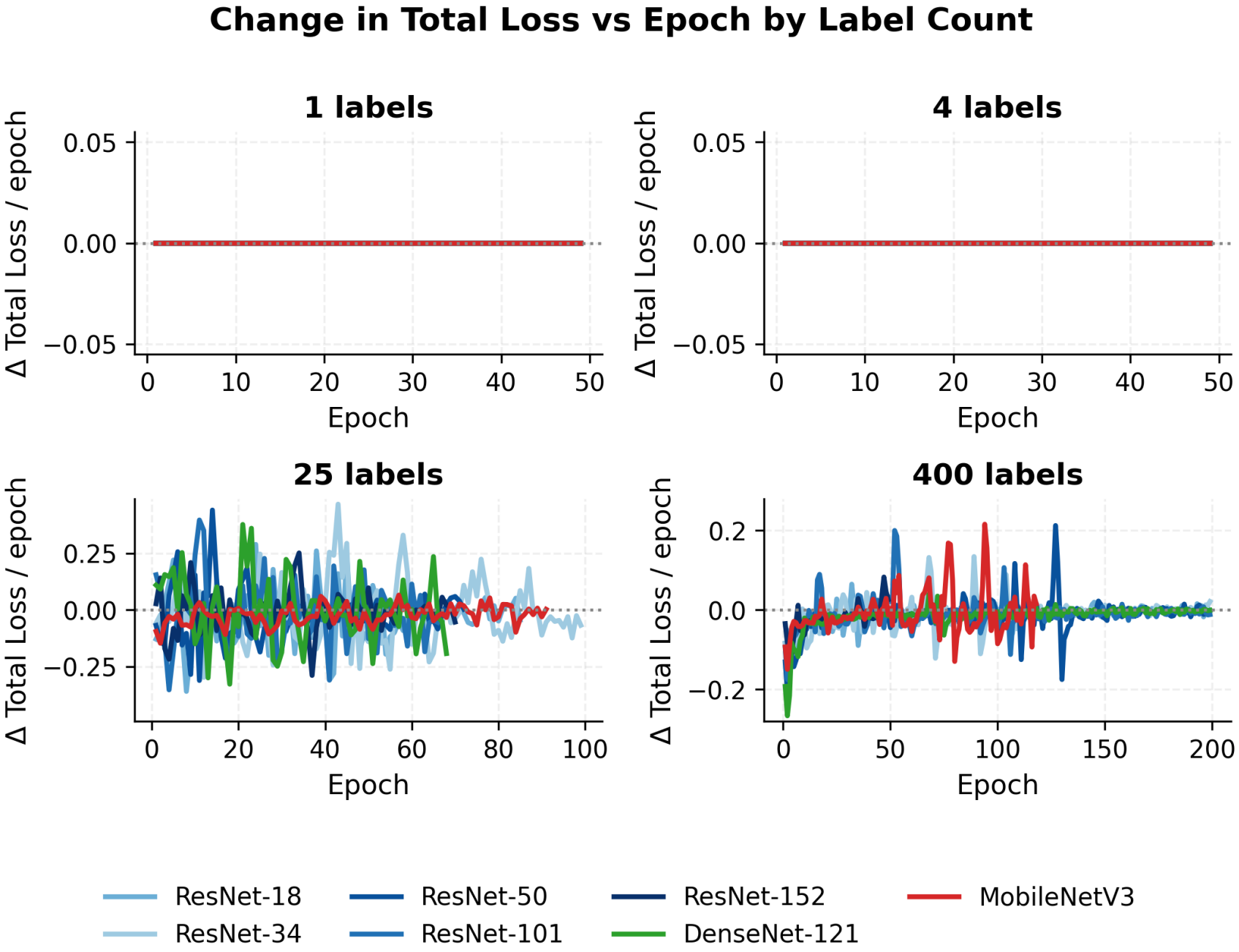
ResNet + Outros

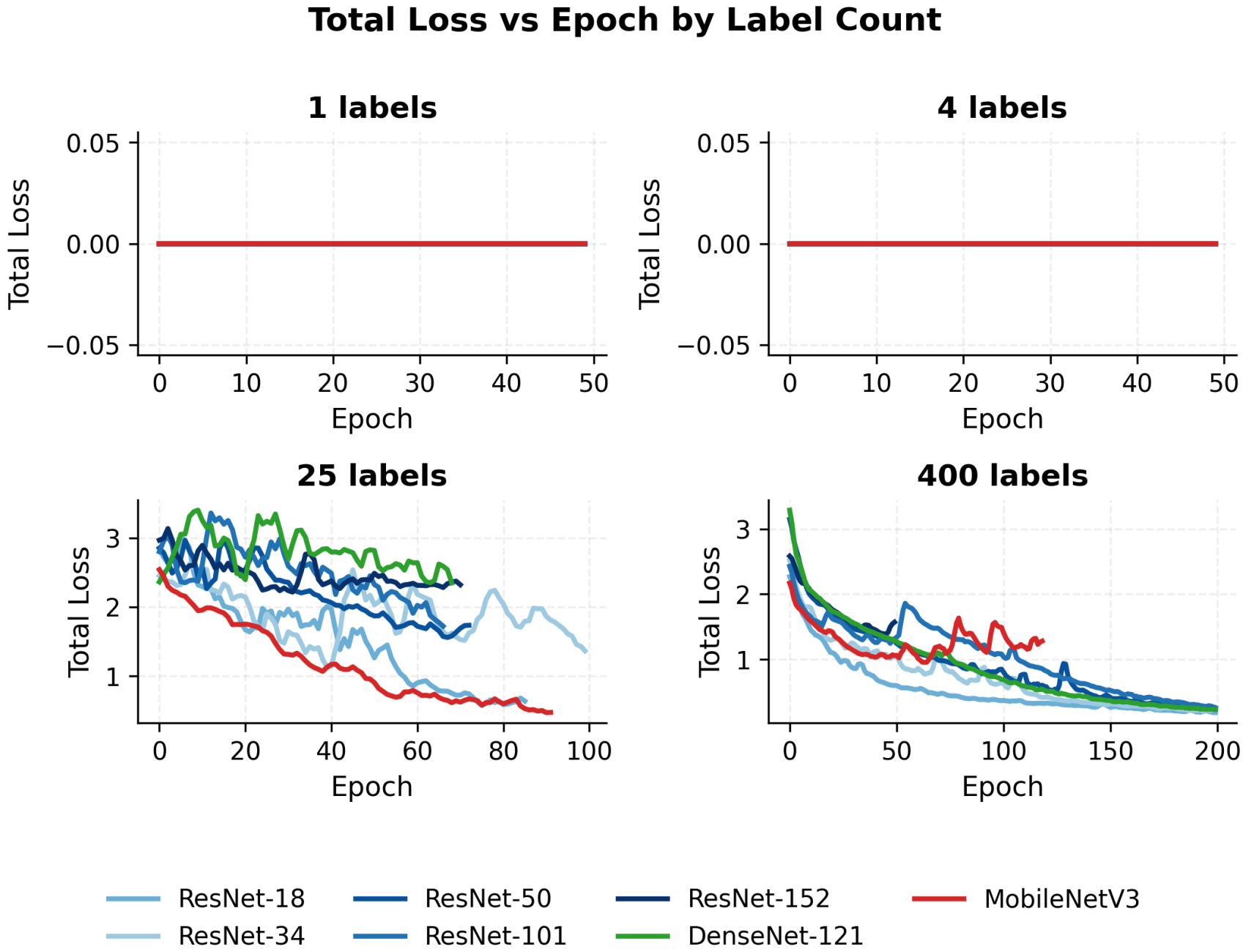

Accuracy vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Accuracy
Epoch
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3

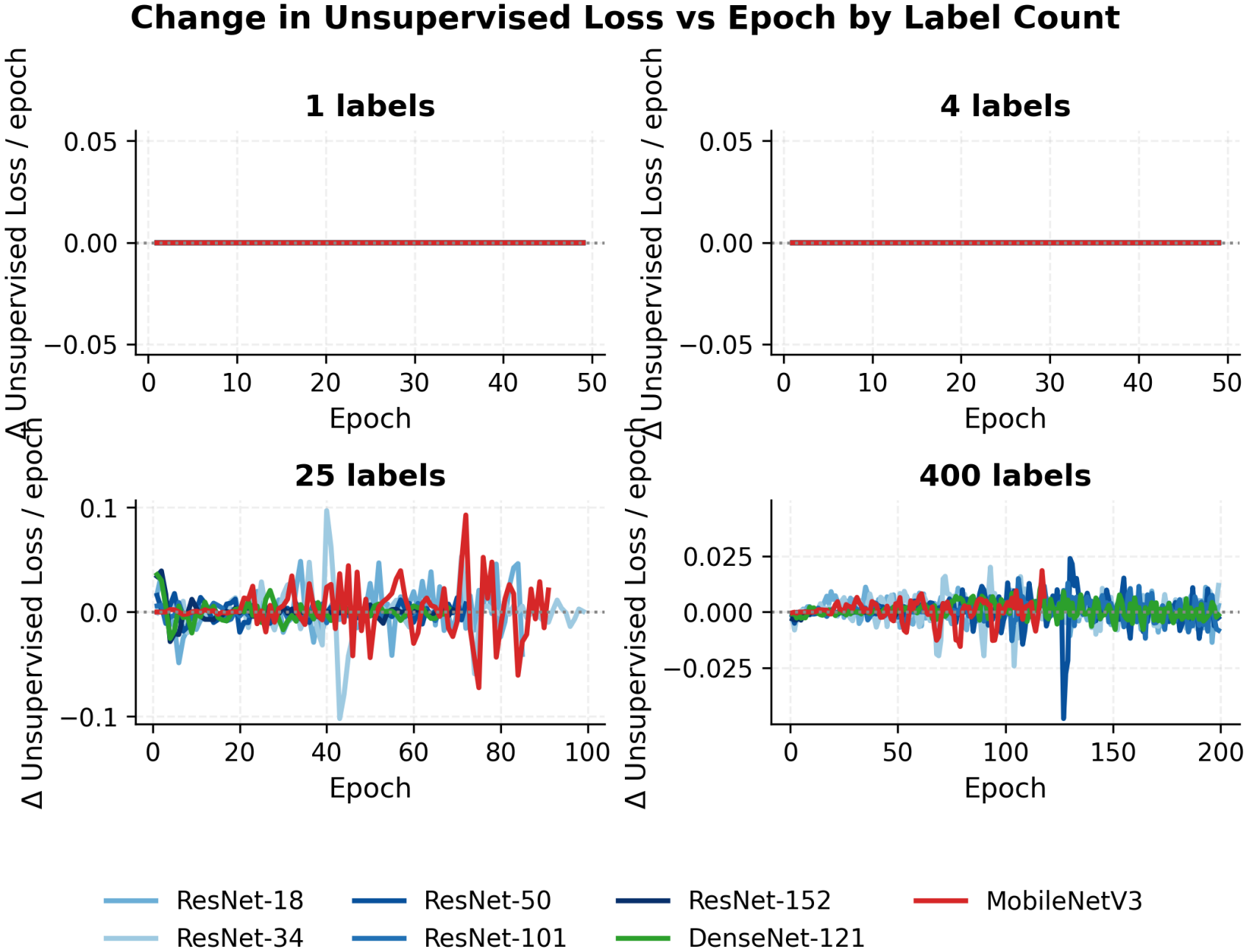

Best Accuracy vs Label Count
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3
Best Accuracy (%)
Number of labeled samples
80
70
60
50
40
30
20
10
1
4
25
400
78.3
73.9
70.3
62.7
53.1
20.4
18.9
14.1
11.4
9.1
7.1

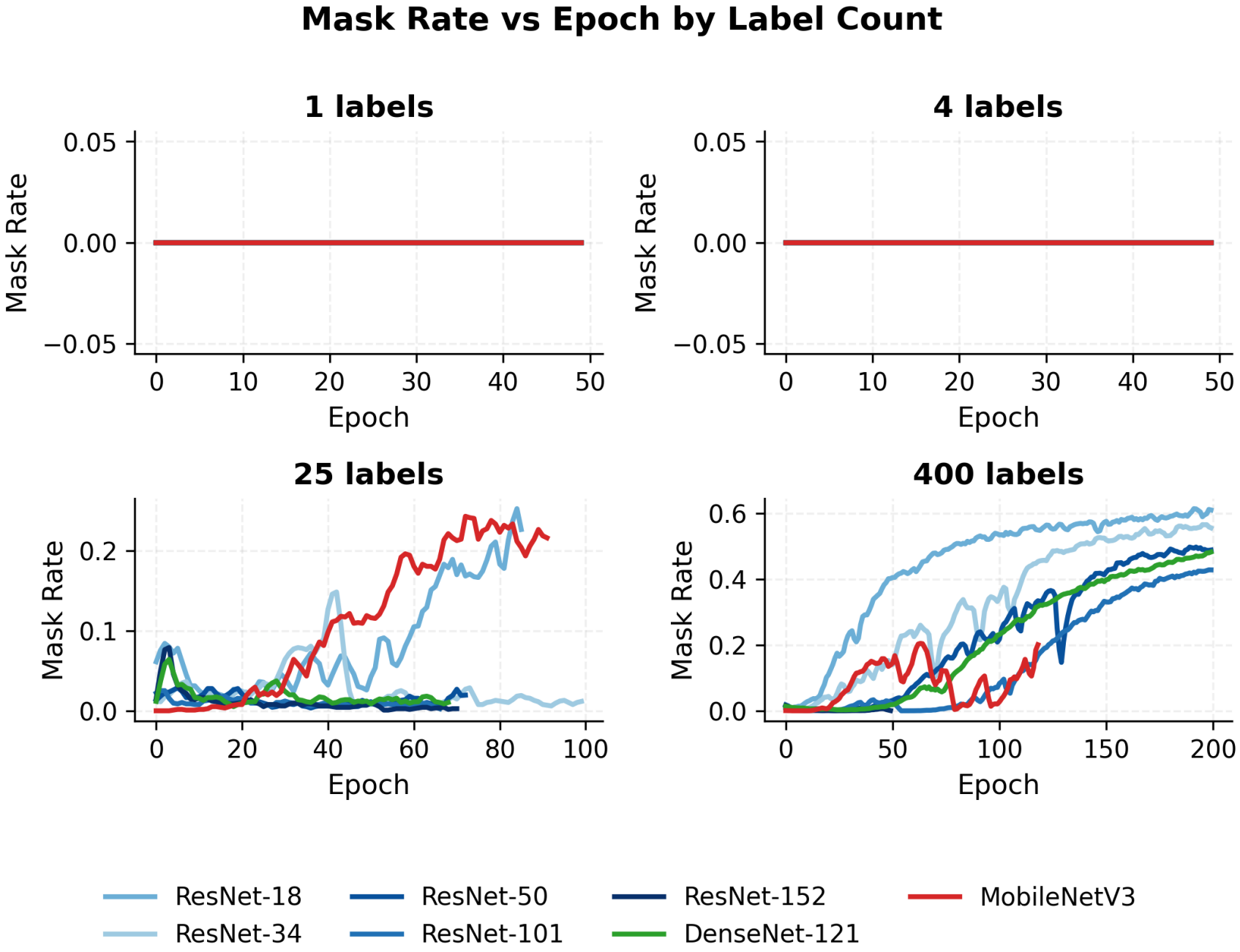

Last-Epoch Accuracy vs Label Count
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3
Last-Epoch Accuracy (%)
Number of labeled samples
80
70
60
50
40
30
20
10
1
4
25
400
78.2
70.3
62.7
38.9
17.5
14.1
17.1


Change in Supervised Loss vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Δ Supervised Loss / epoch
Epoch
0.05
0.00
−0.05
0.5
0.0
−0.5
0.2
−0.2
0
10
20
30
40
50
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3


Change in Total Loss vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Δ Total Loss / epoch
Epoch
0.05
0.00
−0.05
0.25
−0.25
0.2
0.0
−0.2
0
10
20
30
40
50
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3


Total Loss vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Total Loss
Epoch
0.05
0.00
−0.05
0
10
20
30
40
50
1
2
3
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3


Change in Unsupervised Loss vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Δ Unsupervised Loss / epoch
Epoch
0.05
0.00
−0.05
0.1
0.0
−0.1
0.025
0.000
−0.025
0
10
20
30
40
50
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3

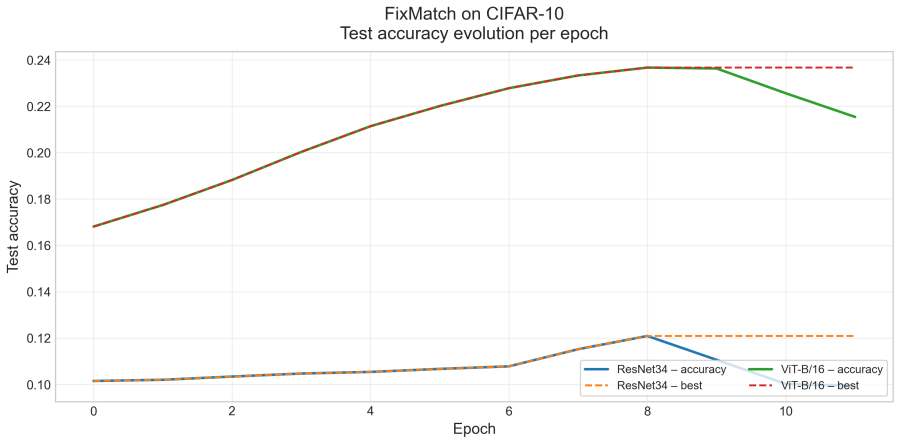

Mask Rate vs Epoch by Label Count
1 labels
4 labels
25 labels
400 labels
Mask Rate
Epoch
0.05
0.00
−0.05
0.2
0.1
0.0
0.6
0.4
0
10
20
30
40
50
60
80
100
150
200
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
DenseNet-121
MobileNetV3

Por que comparar ViT-B/16 e ResNet34?

- **Contraste de paradigmas**: ResNet34 é o backbone convolucional clássico indicado no paper original do FixMatch; ViT-B/16 representa a família transformer com atenção global.
- **Expectativa a priori**: esperávamos que o ViT atingisse acurácia superior após poucas épocas graças à modelagem de longo alcance, enquanto a ResNet serviria como baseline estável e já conhecido.
- **Importância prática**: comparar ambos mostra o quanto ganhos reais vêm de trocar o backbone, mantendo o mesmo pipeline semi-supervisionado (mesmo threshold, EMA, augmentações).
- **Considerações gerais**: ViTs exigem regularização pesada e maior custo computacional; ResNets convergem mais lentamente, mas demandam menos memória. Os gráficos seguintes quantificam esse trade-off.

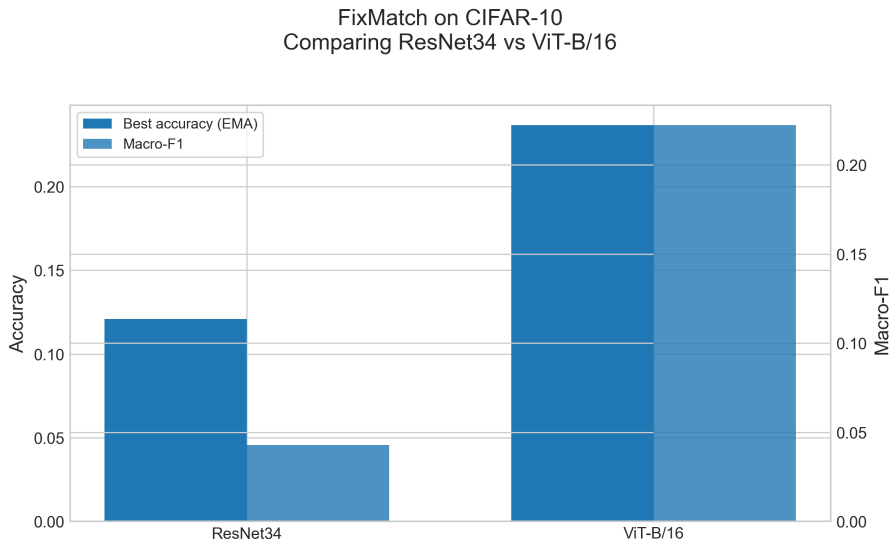
Trajetórias de Acurácia (EMA vs. cru)

- ViT-B/16 já supera 17% de acurácia de teste na época 0 e cruza 20% até a época 4.
- ResNet34 estaciona próximo de 12% mesmo com o EMA, sem ganhos relevantes após a época 8.



Melhor Acurácia vs. Macro-F1

- Melhor acurácia EMA: ViT-B/16 em $23{,}7\%$ vs. ResNet34 em $12{,}1\%$ (gap de $+11{,}6$ pp).
- Macro-F1 repete o cenário: $22{,}2\%$ para ViT-B/16 e apenas $4{,}3\%$ para ResNet34.



Gap de Acurácia ao Longo do Treino

- ViT mantém vantagem em todo o treino, atingindo pico de $+12$ pp perto da época 8.
- Nenhuma época apresenta liderança da ResNet, indicando benefício estrutural do backbone transformer.
- O gap diminui levemente após a época 10, sugerindo convergência sem cruzamento das curvas.