

FixMatch no CIFAR-10: implementação, experimentos e análise

Anderson Falcão

Leonardo Alexandre

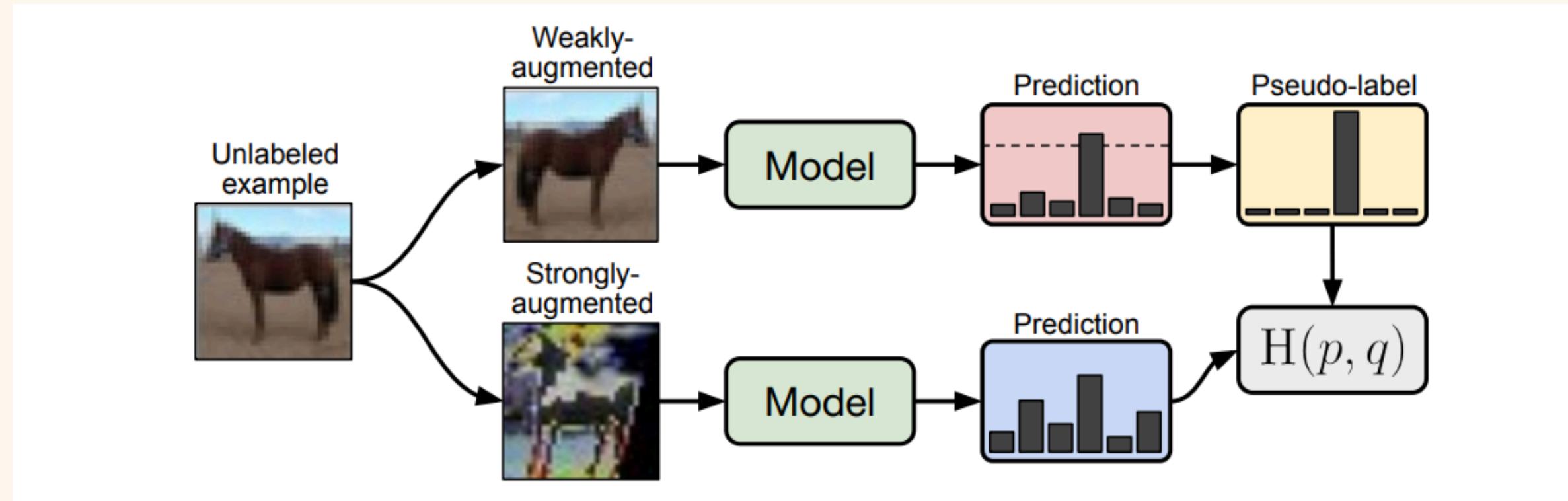
Ramyro Corrêa

Introdução

- FixMatch reduz a necessidade de rótulos no CIFAR-10.
- Com poucos rótulos por classe, é possível alcançar boas acurácia alinhando pseudo-rótulos confiáveis com fortes aumentações.
- Pontos-chave: algoritmo semi-supervisionado (supervisionado + consistência), pseudo-rótulos só entram quando a confiança $\geq \tau$.
- Avaliamos 1, 4, 25, 70 e 400 rótulos/classe.

Introdução

- x (não rotulado) \rightarrow weak aug $\rightarrow p(y|x^{\text{weak}})$
- Se $\max p \geq \tau \Rightarrow$ pseudo-rótulo $\hat{y} = \text{argmax } p$
- Mesma x com strong aug \rightarrow treinar para prever \hat{y}
- Em paralelo: treino supervisionado nas amostras rotuladas
- Intuição: A versão fraca fornece um rótulo confiável; a versão forte força robustez.



Loss Híbrida

$$\text{Loss} = \text{Loss_s} + \lambda_u \cdot \text{Loss_ns}$$

$$\ell_s = \frac{1}{B} \sum_{b=1}^B H(p_b, p_m(y | \alpha(x_b)))$$

$$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau) H(\hat{q}_b, p_m(y | \mathcal{A}(u_b)))$$

Ideia: unir um “sinal limpo” (rótulos verdadeiros) com um “sinal barato” (pseudo-rótulos confiáveis) para empurrar o modelo a ser consistente mesmo sob aumentações fortes, de modo que rótulos reais ancam o modelo; pseudo-rótulos espalham esse aprendizado para o mar de dados não rotulados, desde que sejam confiáveis.

Implementação

- Modelo base: ResNet-18
- Dados: CIFAR-10 (divisão em rotulados vs. não rotulados)
- Augmentações: flip/crop padrão para weak e RandAugment para strong.
- Treino: SGD+Nesterov Momentum, scheduler, weight decay, EMA
- Logging: acurácia, losses, acceptance rate de pseudo-rótulos

```
weak_transform = transforms.Compose([
    transforms.RandomHorizontalFlip(),
    transforms.RandomCrop(32, padding=4, padding_mode='reflect'),
    transforms.ToTensor(),
    transforms.Normalize(MEAN, STD),
])

strong_transform = transforms.Compose([
    transforms.RandomHorizontalFlip(),
    transforms.RandomCrop(32, padding=4, padding_mode='reflect'),
    RandAugment(num_ops=2, magnitude=10),
    transforms.ToTensor(),
    transforms.Normalize(MEAN, STD),
])

test_transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize(MEAN, STD),
])
```

```
class FixMatchLoss(nn.Module):
    def __init__(self, tau=0.95, lambda_u=1.0):
        super().__init__()
        self.tau = tau
        self.lambda_u = lambda_u
        self.ce = nn.CrossEntropyLoss()

    @torch.no_grad()
    def pseudo(self, logits_weak):
        p = torch.softmax(logits_weak, dim=1)
        conf, yhat = p.max(dim=1)
        mask = (conf >= self.tau).float()
        return yhat, mask

    def forward(self, logits_sup, y_sup, logits_w, logits_s):
        Ls = self.ce(logits_sup, y_sup)
        yhat, mask = self.pseudo(logits_w)
        Lu_all = F.cross_entropy(logits_s, yhat, reduction='none')
        Lu = (Lu_all * mask).sum() / (mask.sum().clamp_min(1.0))
        return Ls + self.lambda_u*Lu, (Ls.item(), Lu.item(), mask.mean().item())
```

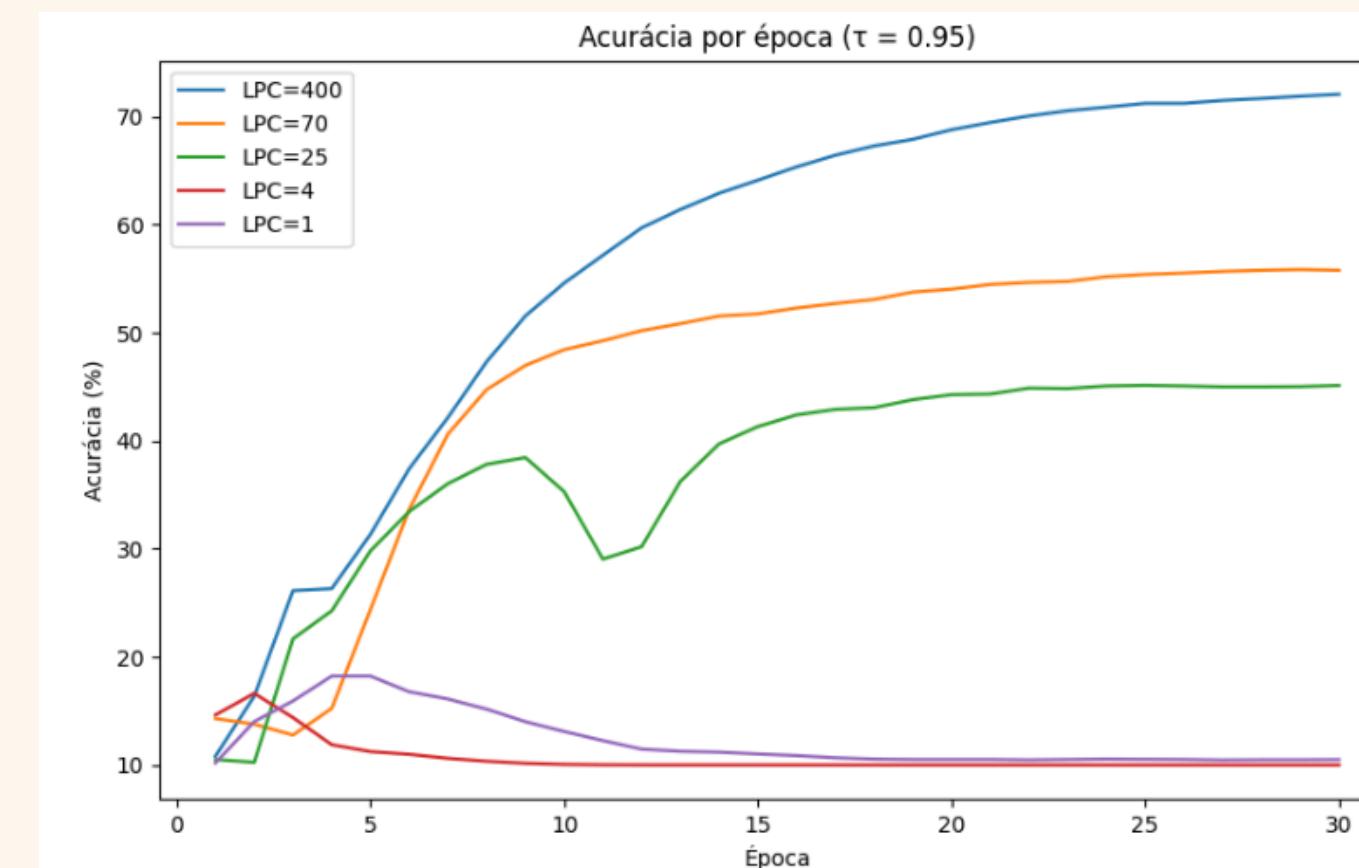
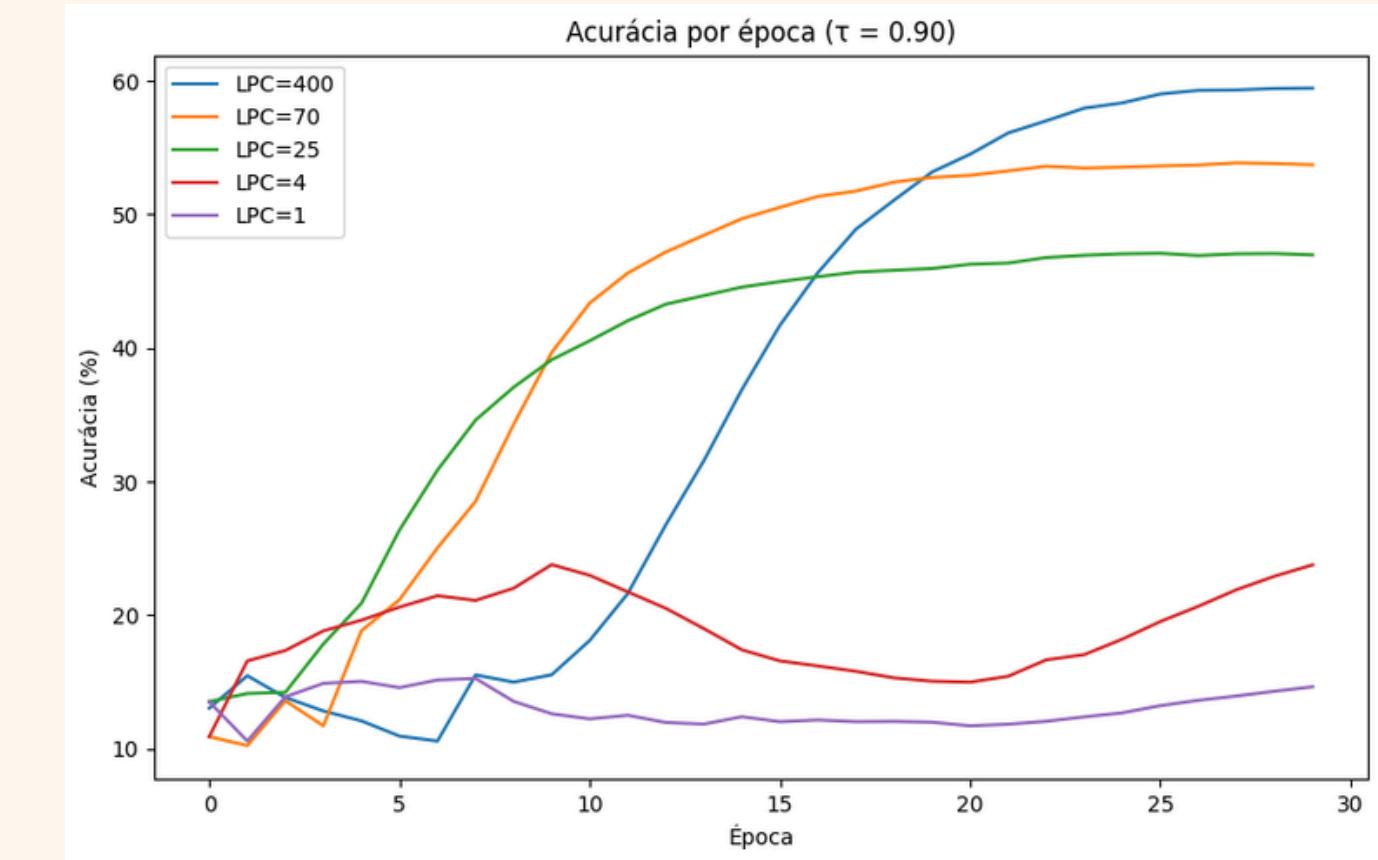
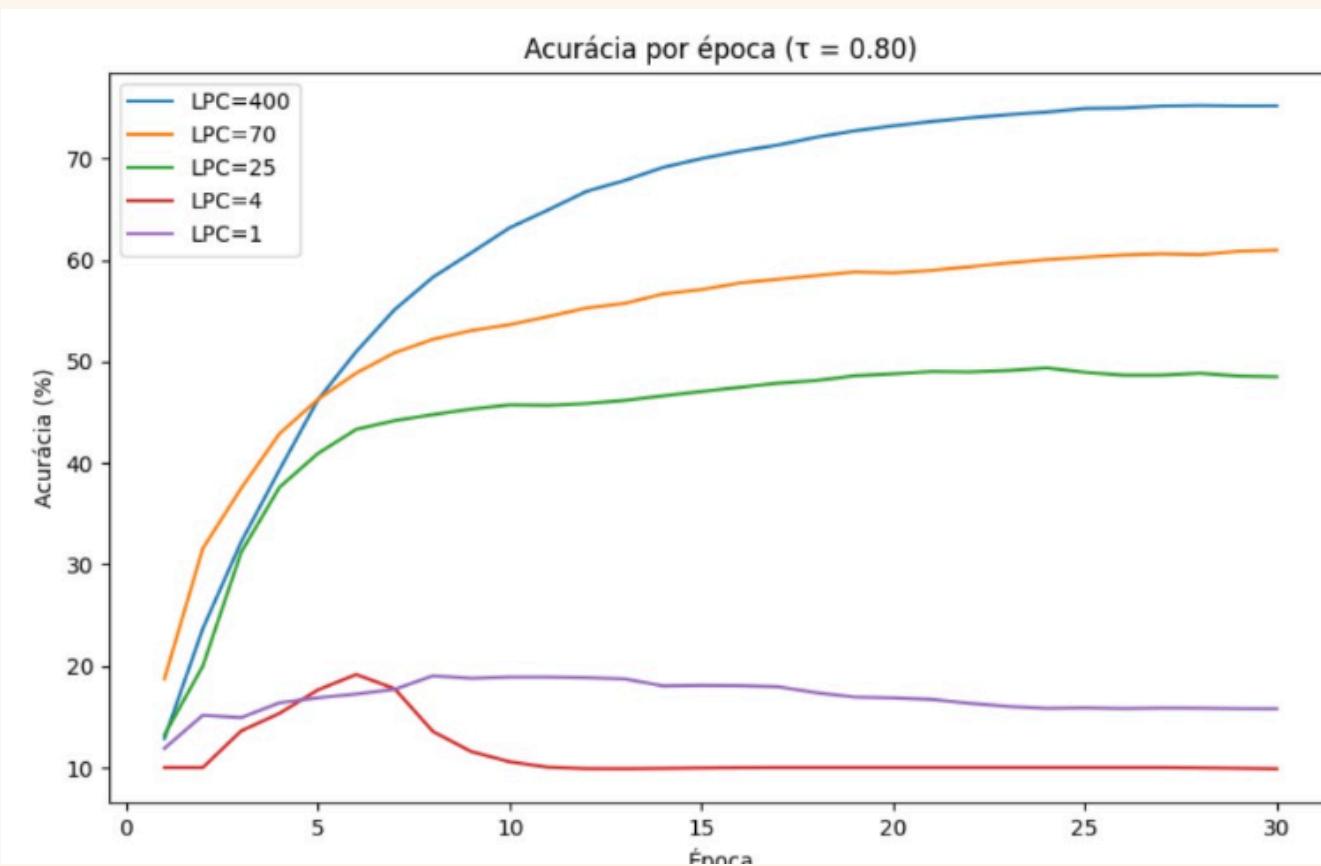
Experimentos

- Cenários: 1, 4, 25, 70 e 400 rótulos por classe (total: 10, 40, 250, 700, 4.000 dados rotulados no total), em um universo de 10.000 dados.
- Amostragem: estratificada por classe (mesma quantidade por classe)
- $\tau = 0.80, 0.9, 0.95$
- Configuração: $\mu \approx 7$; $\lambda_u = 1$; $LR = 0.03$
- Métricas: acurácia de teste (%), loss de teste e acceptance rate (fração de pseudo-rótulos aceitos).

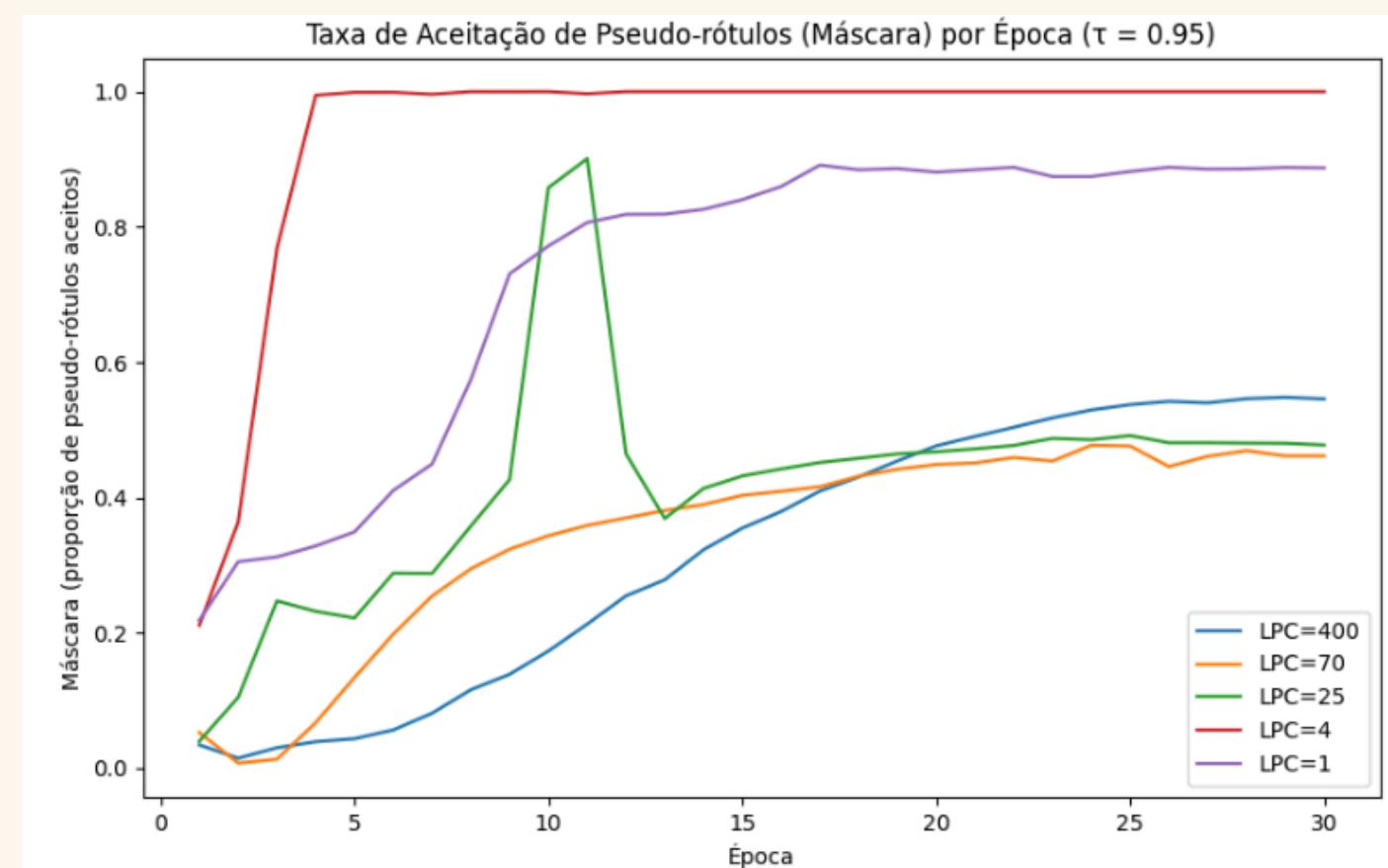
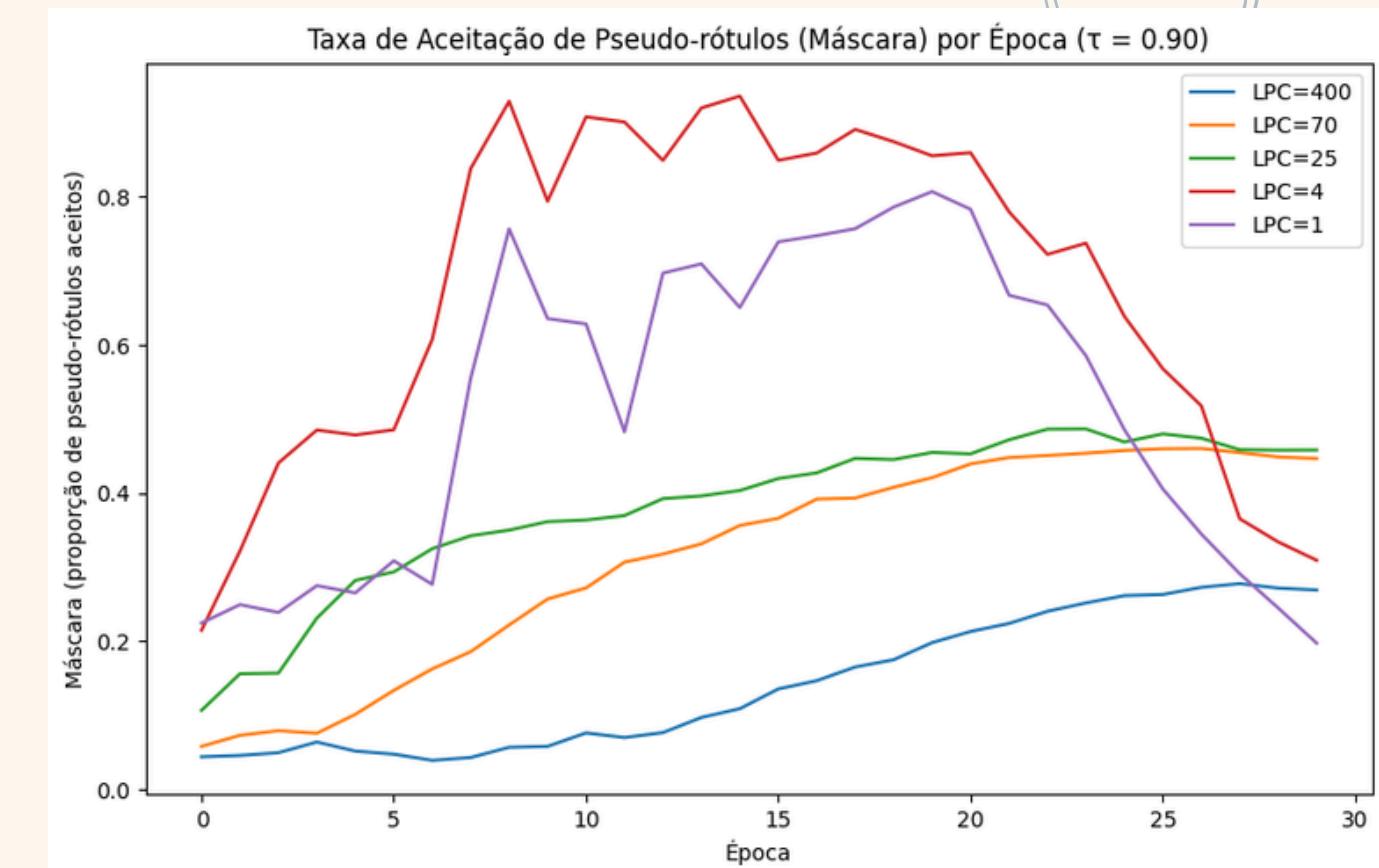
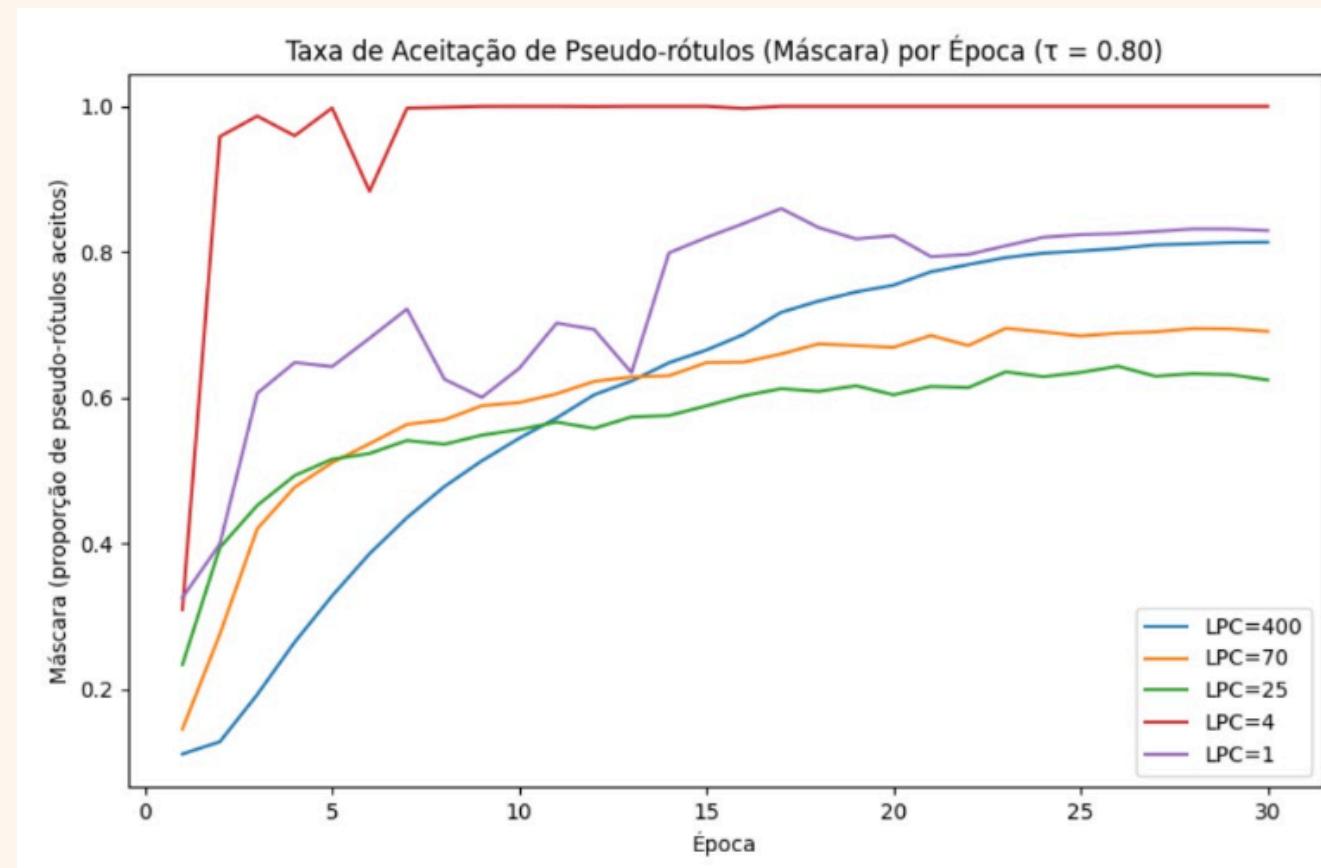
Resultados

- Qualidade dos pseudo-rótulos
- Impacto do threshold de confiança
- Impacto do augmentation e loss híbrida

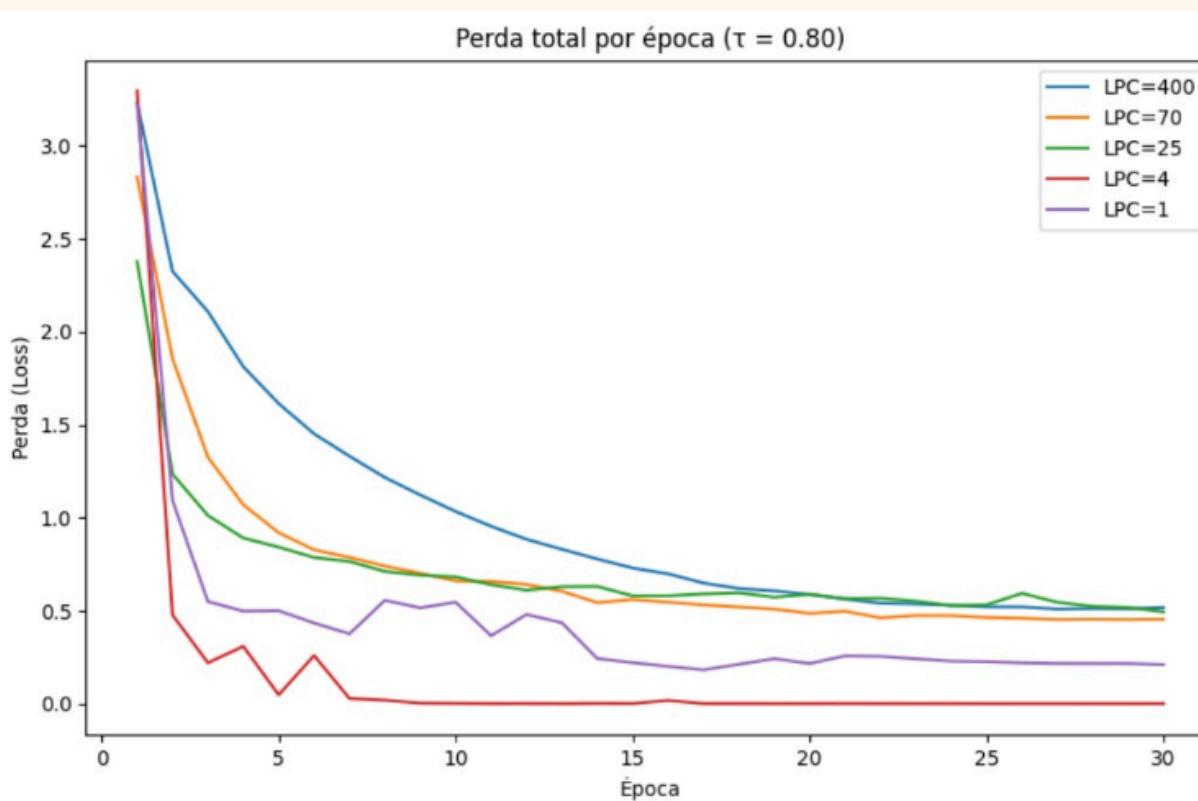
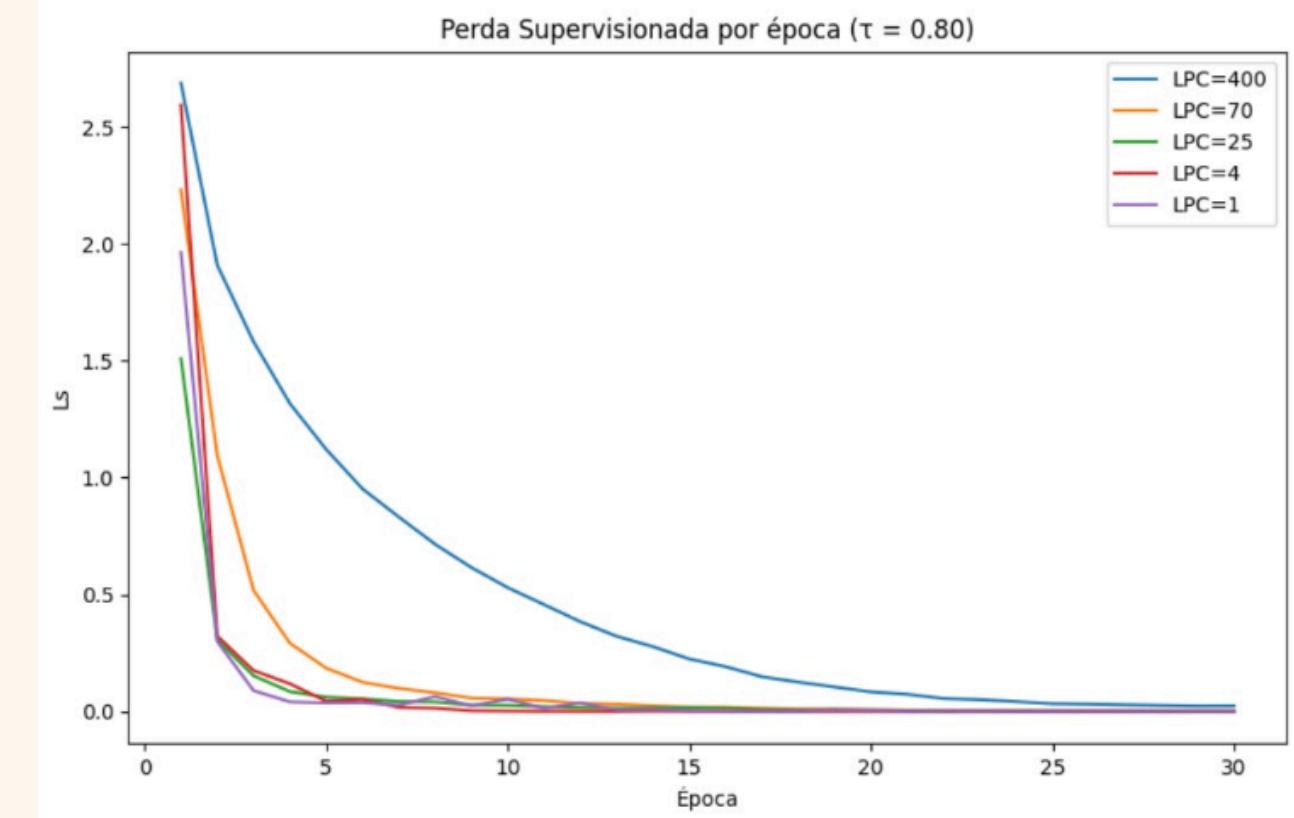
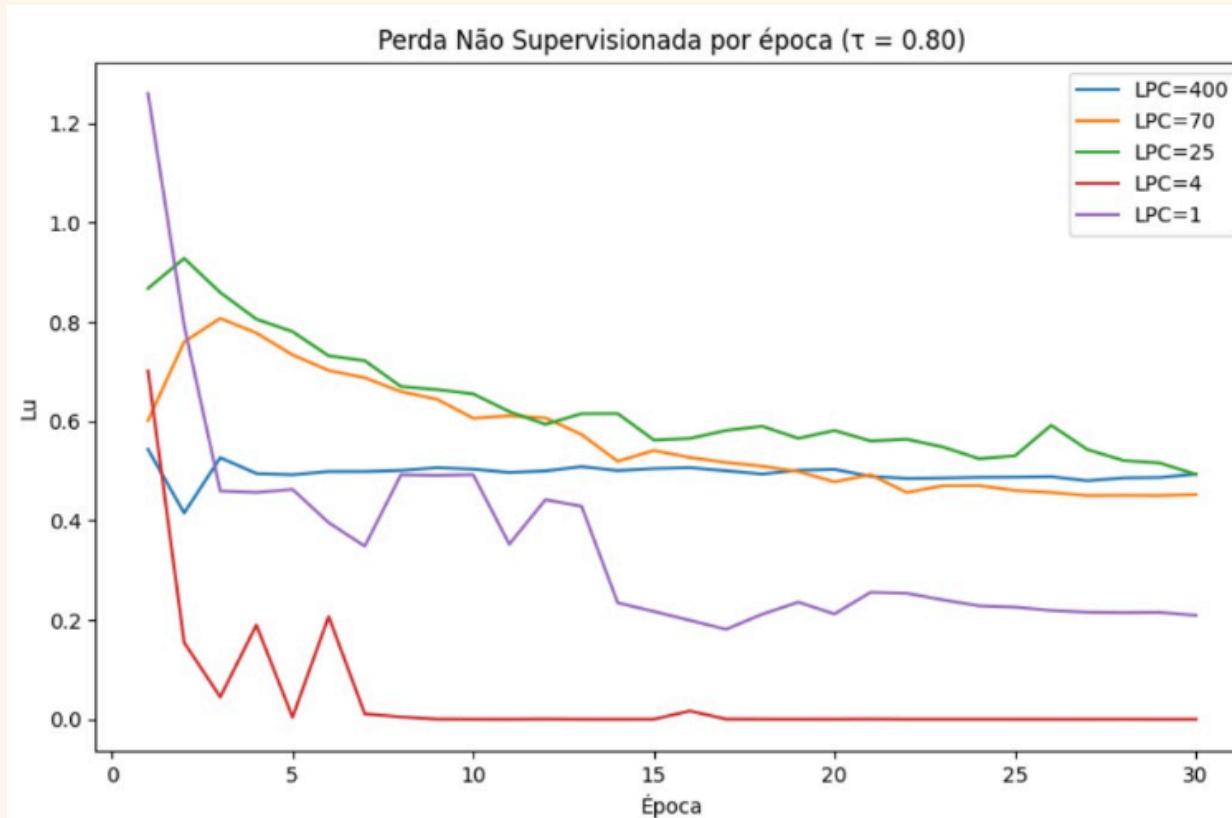
Impacto do threshold de confiança



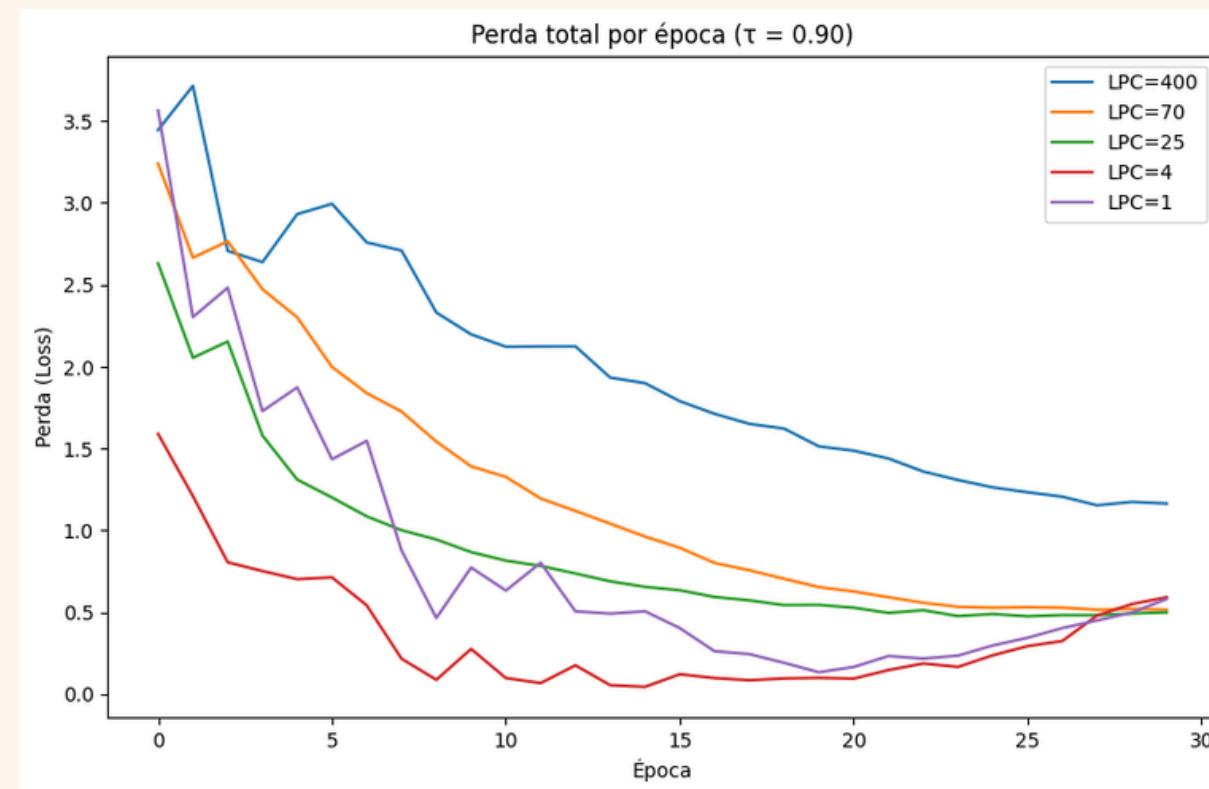
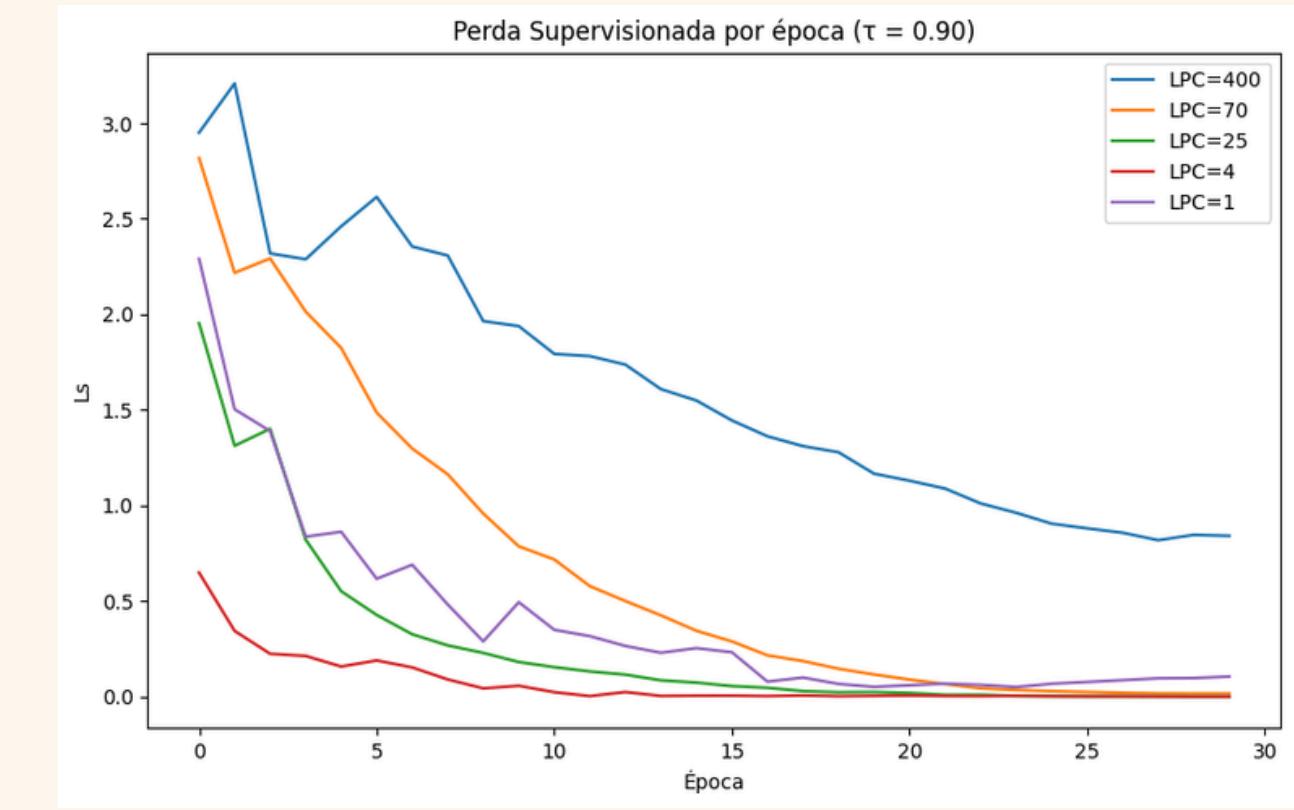
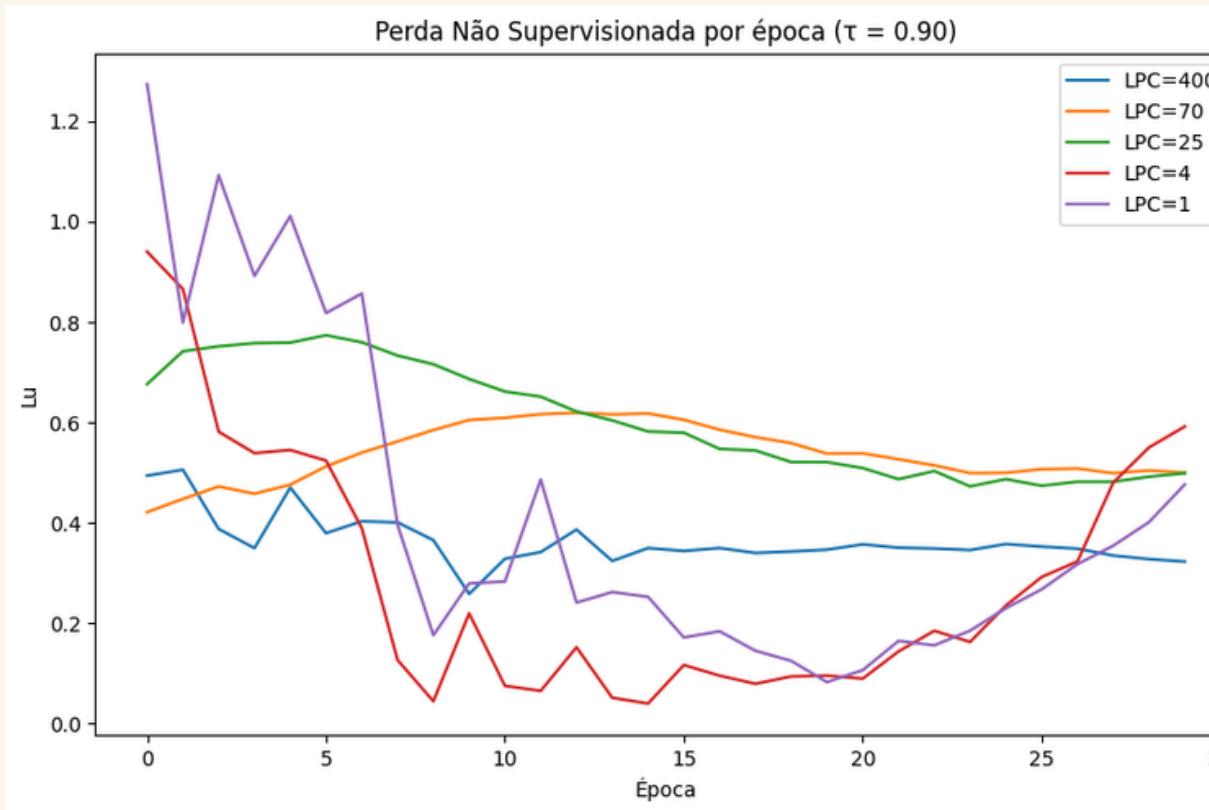
Qualidade dos pseudo-rótulos



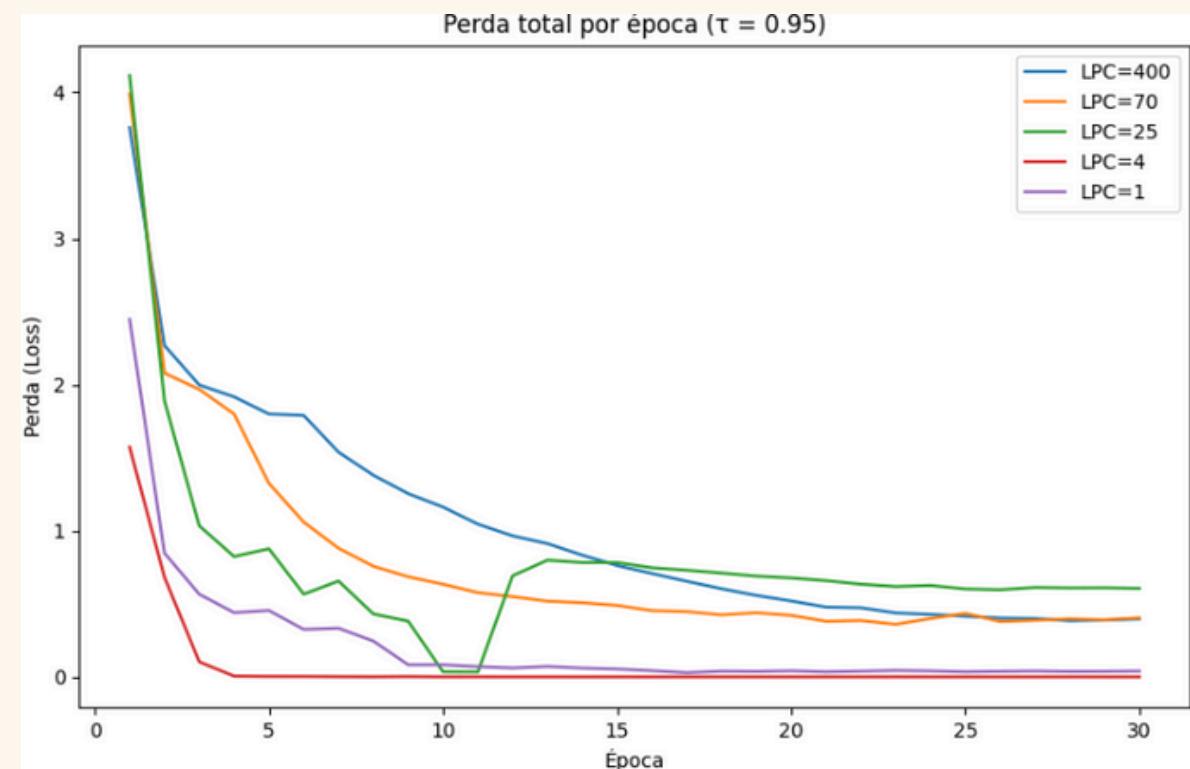
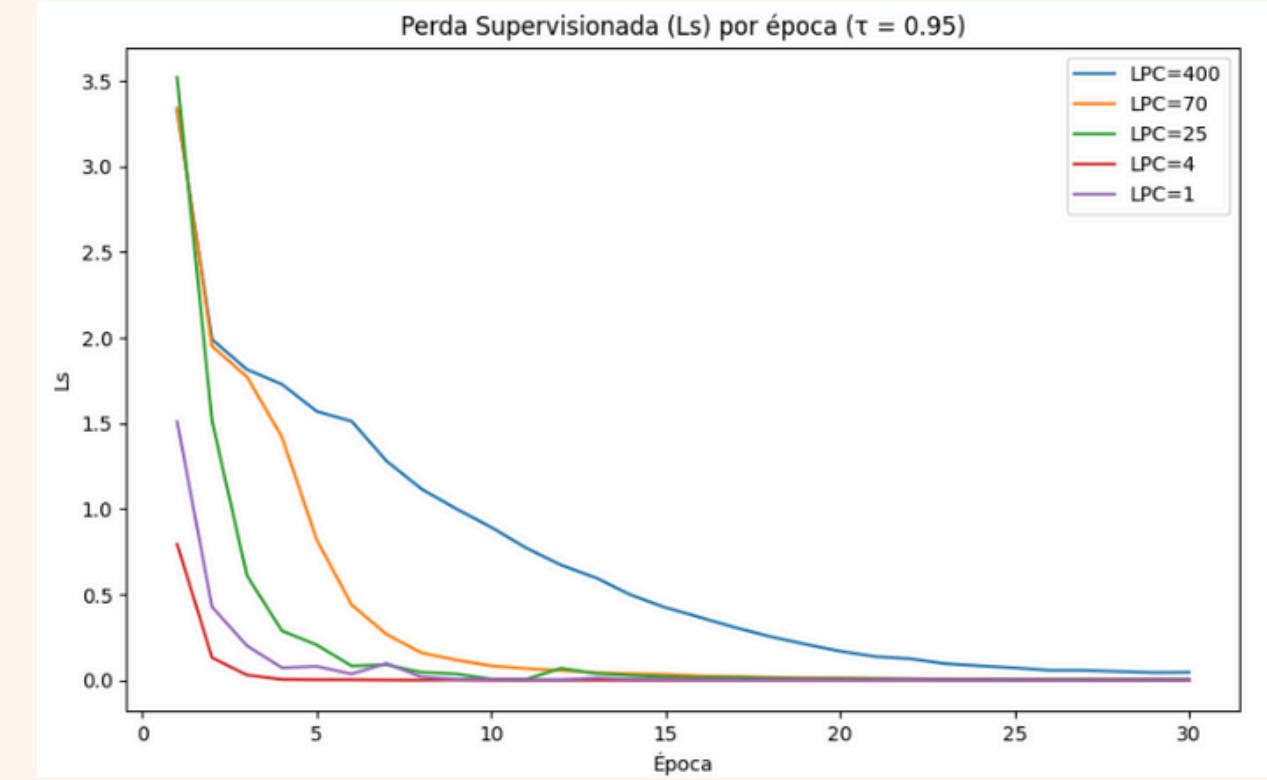
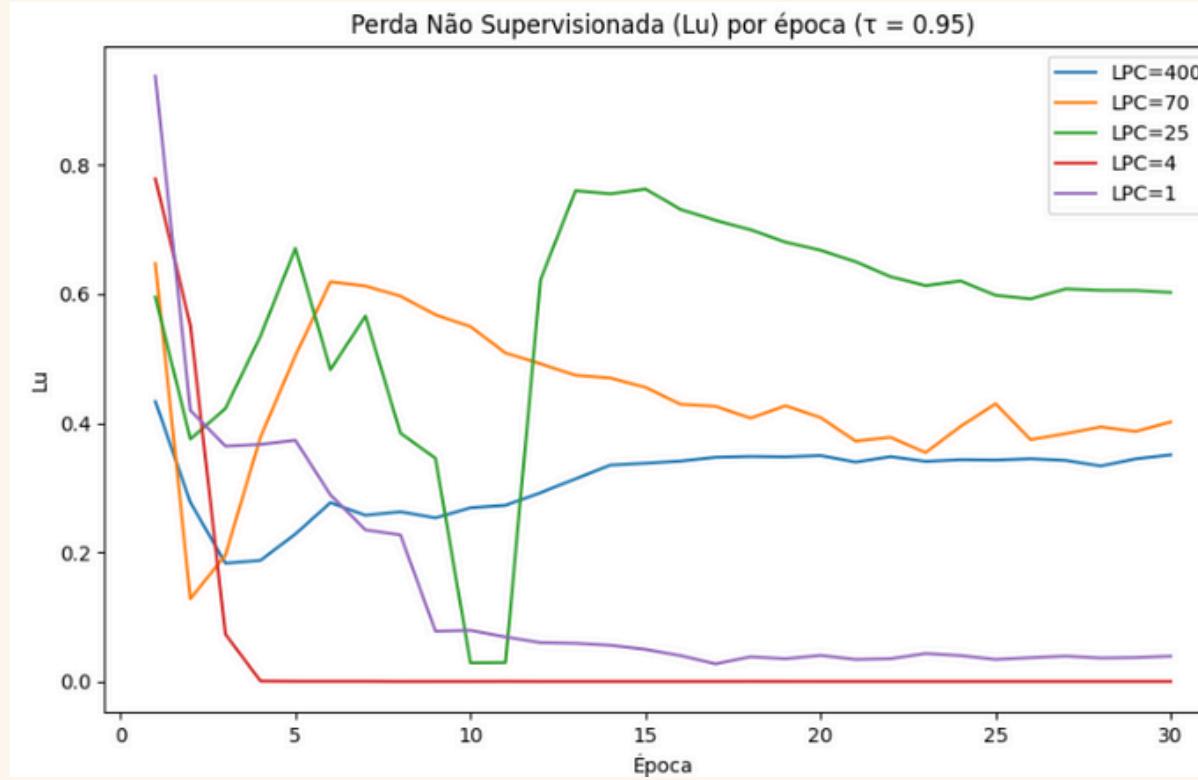
Impacto do augmentation e loss híbrida



Impacto do augmentation e loss híbrida



Impacto do augmentation e loss híbrida



Obrigado!