

Deep Learning

CLIP: *Contrastive Language-Image Pre-training*

Lucas Silveira Kupssinskü

Machine Learning Theory and Applications Laboratory
PUCRS

agosto de 2026

Visão Geral

1. **Motivação e Contexto**
2. **Aprendizado Contrastivo**
3. **Arquitetura do CLIP**
4. **Dataset e Treinamento**
5. **Avaliação e Uso**
6. **Limitações e Sucessores**

Visão Geral

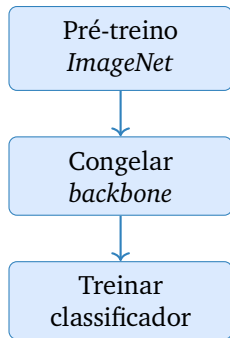
- 1. Motivação e Contexto**
2. Aprendizado Contrastivo
3. Arquitetura do CLIP
4. Dataset e Treinamento
5. Avaliação e Uso
6. Limitações e Sucessores

Antes do CLIP: Transferência por Pré-treino Supervisionado

Até 2020, o caminho dominante em visão computacional era:

- Pré-treinar uma CNN em ImageNet (1.28M imagens, 1000 classes)
- Reaproveitar as camadas convolucionais como *feature extractor*
- Treinar apenas o classificador final (*linear probe*)
- Eventualmente, *fine-tuning* de mais camadas com taxa de aprendizado pequena

Receita robusta, porém com dois custos altos: rótulos manuais e classes fixas.



Visão Computacional vs. PLN em 2020

PLN

- Pré-treino *auto-supervisionado* massivo (BERT, GPT-2, GPT-3)
- Bilhões de tokens de texto bruto da web
- Sem necessidade de rótulos humanos

Visão Computacional

- Ainda dependente de ImageNet e datasets curados
- Rotulação manual cara e lenta
- *Self-supervised* (SimCLR, MoCo) começava a aparecer, mas restrito a uma modalidade

Pergunta: podemos importar para visão a receita “coleta web + pré-treino sem rótulos”?

O Sonho do *Zero-Shot* em Visão

- Modelos supervisionados só reconhecem o conjunto fixo de classes em que foram treinados
- Adicionar uma classe nova exige coletar exemplos rotulados e re-treinar
- *Zero-shot*: classificar uma classe nunca vista no treino, descrita apenas por linguagem natural

Definição operacional

Dado um modelo treinado e uma nova classe descrita por uma string (“a photo of a tapir”), prever se uma imagem pertence a essa classe **sem nenhum exemplo de tapir no treino**.

Para isso, precisamos de um espaço onde **imagem** e **texto** convivam.

Hipótese-Chave: Legendas da Web São Supervisão Suficiente

Cada imagem na web tende a vir acompanhada de um texto:

- alt-text de HTML
- Legenda de Wikipedia, Instagram, Flickr
- Texto adjacente em artigos de notícias

Esses pares (imagem, texto) são **ruidosos** e **abundantes**: anti-curadoria como filosofia de coleta.

Resta projetar uma função de custo que aproveite esse sinal fraco em escala massiva.



“a fluffy puppy
in the garden”

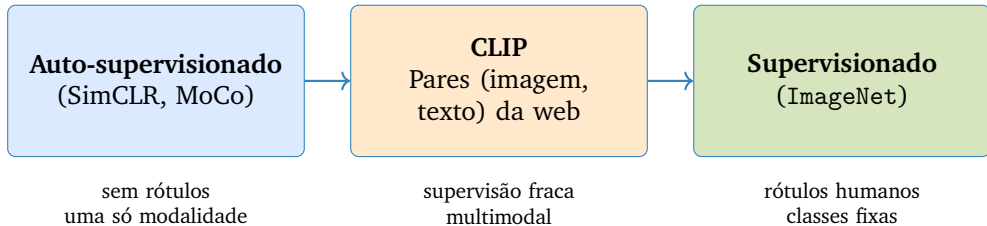


“Eiffel Tower in Paris”



“Earth from Apollo 17”

Onde CLIP se Encaixa no Espectro de Supervisão



CLIP herda *escala* do auto-supervisionado e *semântica linguística* do supervisionado.

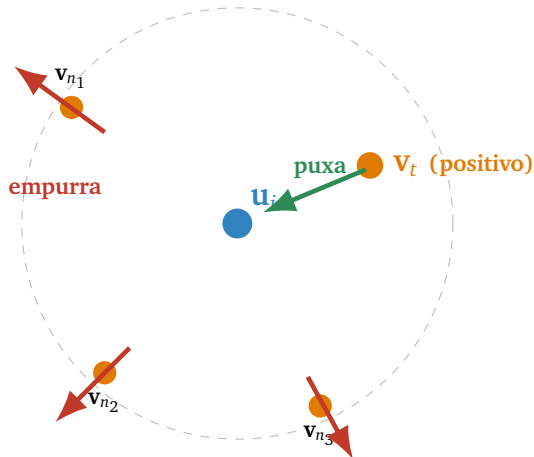
Visão Geral

1. Motivação e Contexto
- 2. Aprendizado Contrastivo**
3. Arquitetura do CLIP
4. Dataset e Treinamento
5. Avaliação e Uso
6. Limitações e Sucessores

Intuição: Aproximar Pares Positivos, Afastar Negativos

Para cada par **positivo** (imagem i , texto t), com *embeddings* u_i (imagem) e v_t (texto) projetados no **mesmo espaço**:

- **Puxar** u_i e v_t para perto no espaço de *embeddings*
- **Empurrar** u_i para longe de **todos os outros textos** do batch
- Tanto de texto para imagem quanto de imagem para texto



Os “empurrões” e “puxões” são gradientes da função de custo *InfoNCE*.

Similaridade do Cosseno

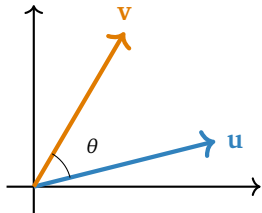
Definição

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \cos \theta$$

Em CLIP, **normalizamos $\mathbf{u}$ e $\mathbf{v}$ para norma 1** antes de computar a similaridade:

$$\tilde{\mathbf{u}} = \mathbf{u} / \|\mathbf{u}\|, \quad \tilde{\mathbf{v}} = \mathbf{v} / \|\mathbf{v}\|$$

Assim $\text{sim}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \tilde{\mathbf{u}}^\top \tilde{\mathbf{v}} \in [-1, 1]$.



A Perda *InfoNCE*¹

Em um *batch* de N pares $\{(\mathbf{u}_i, \mathbf{v}_i)\}_{i=1}^N$ (já normalizados), formamos $S \in \mathbb{R}^{N \times N}$ com $S_{ij} = \mathbf{u}_i^\top \mathbf{v}_j$. Aqui $\tau > 0$ é a *temperatura* (próximo slide): controle de “afiamento” da *softmax*.

Loss simétrica de CLIP

Cada direção é uma *softmax* com alvo na diagonal ($i \rightarrow t$: **linha** i ; $t \rightarrow i$: **coluna** j):

$$\mathcal{L}_{i \rightarrow t} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(S_{ii}/\tau)}{\sum_{j=1}^N \exp(S_{ij}/\tau)}$$

$$\mathcal{L}_{t \rightarrow i} = -\frac{1}{N} \sum_{j=1}^N \log \frac{\exp(S_{jj}/\tau)}{\sum_{i=1}^N \exp(S_{ij}/\tau)}, \quad \mathcal{L}_{\text{CLIP}} = \frac{1}{2} (\mathcal{L}_{i \rightarrow t} + \mathcal{L}_{t \rightarrow i})$$

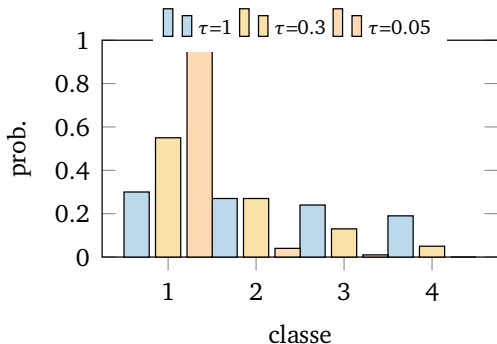
InfoNCE = **cross-entropy** (*softmax* + $-\log$) com a diagonal como rótulo: classificação de N **vias no *batch***.

¹Aaron van den Oord, Yazhe Li e Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding”. Em: [arXiv preprint arXiv:1807.03748](https://arxiv.org/abs/1807.03748) (2018).

Por Que a Temperatura τ Importa

- τ pequeno $\Rightarrow$ *softmax* mais “afiada” (concentra probabilidade no maior *logit*)
- τ grande $\Rightarrow$ distribuição mais uniforme, gradientes menos focados
- Em CLIP, τ é **aprendido** (parâmetro $\log(1/\tau)$, inicializado em $\log 14$, recortado em $\log 100$)

Valor convergido típico: $1/\tau \approx 100$, ou seja $\tau \approx 0.01$.



Exemplo Trabalhado I: Matriz de Similaridade

Batch com $N = 2$. *Embeddings* já normalizados em $\mathbb{R}^3$:

$$\mathbf{u}_1 = \begin{pmatrix} 0.8 \\ 0.6 \\ 0 \end{pmatrix}, \mathbf{u}_2 = \begin{pmatrix} 0 \\ 0.8 \\ 0.6 \end{pmatrix}, \mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 0 \\ 0.6 \\ 0.8 \end{pmatrix}$$

Cada $S_{ij} = \mathbf{u}_i^\top \mathbf{v}_j$:

- $S_{11} = 0.8 \cdot 1 + 0.6 \cdot 0 + 0 = \mathbf{0.80}$ (positivo)
- $S_{12} = 0 + 0.6 \cdot 0.6 + 0 = \mathbf{0.36}$ (negativo)
- $S_{21} = 0 + 0.8 \cdot 0 + 0.6 \cdot 0 = \mathbf{0.00}$ (negativo)
- $S_{22} = 0 + 0.8 \cdot 0.6 + 0.6 \cdot 0.8 = \mathbf{0.96}$ (positivo)

$$S = \begin{pmatrix} \mathbf{0.80} & 0.36 \\ 0.00 & \mathbf{0.96} \end{pmatrix}$$

Diagonal = pares positivos; fora da diagonal = negativos. Note que S **não é simétrica**.

Exemplo Trabalhado II: *InfoNCE* com τ Diferente

Aplicamos *softmax* linha a linha em S/τ . A perda usa a probabilidade do par correto (a **diagonal**):

τ	S/τ linha 1	softmax	S/τ linha 2	softmax	$\mathcal{L}_{i \rightarrow t}$
1.0	(0.80, 0.36)	(0.608, 0.392)	(0.00, 0.96)	(0.277, 0.723)	0.411
0.1	(8.0, 3.6)	(0.988, 0.012)	(0.0, 9.6)	(≈ 0 , ≈ 1)	0.006
0.01	(80, 36)	(≈ 1 , ≈ 0)	(0, 96)	(≈ 0 , ≈ 1)	≈ 0

Para $\tau=1$: $\mathcal{L}_{i \rightarrow t} = -\frac{1}{2}(\log 0.608 + \log 0.723) = 0.411$. A outra direção usa a *softmax* por **coluna**: $\mathcal{L}_{t \rightarrow i} = -\frac{1}{2}(\log 0.690 + \log 0.646) = 0.404$ (**diferente!**), logo
 $\mathcal{L}_{\text{CLIP}} = \frac{1}{2}(0.411 + 0.404) = 0.408$.

- Diminuir τ **recompensa fortemente** positivos já bem separados.
- Mas amplifica ruído: um negativo enganoso vira gradiente enorme.
- Daí a temperatura aprendível com *clipping* em $\log 100$.

Negativos Vêm de Graça do *Batch*

Em um batch com N pares positivos, cada imagem “vê” $N-1$ textos negativos (e vice-versa). Total:

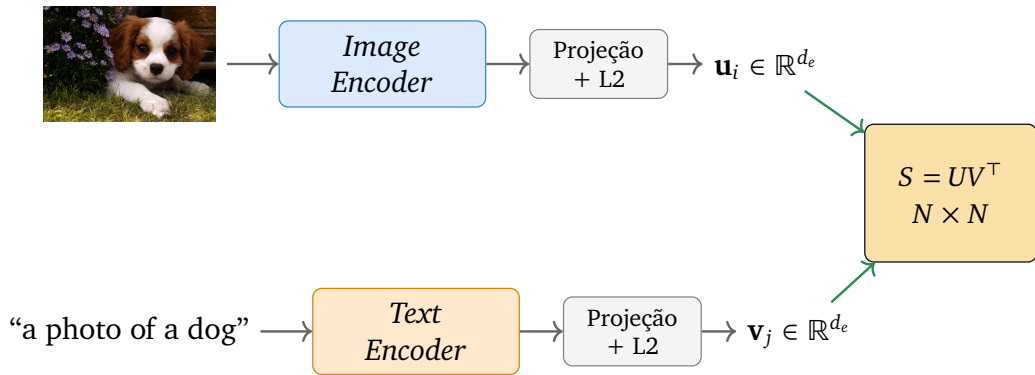
$$\underbrace{N}_{\text{positivos}} + \underbrace{N(N-1)}_{\text{negativos}} = N^2 \text{ comparações na matriz } S$$

N	positivos	negativos
32	32	992
256	256	65 280
8 192	8 192	67 100 672
32 768	32 768	$\approx 1.07 \times 10^9$

Visão Geral

1. Motivação e Contexto
2. Aprendizado Contrastivo
- 3. Arquitetura do CLIP**
4. Dataset e Treinamento
5. Avaliação e Uso
6. Limitações e Sucessores

Arquitetura: Dois *Encoders*, Uma Matriz²



Treinamento conjunto: gradiente flui pelos dois *encoders* a partir de $\mathcal{L}_{\text{CLIP}}$.

²Alec Radford et al. "Learning Transferable Visual Models from Natural Language Supervision". Em: [ICML](#). 2021, pp. 8748–8763.

Encoder de Imagem: Duas Famílias

Variantes ResNet

- ResNet-50, 101, 50x4, 50x16, 50x64
- Modificação: *attention pooling* no lugar do *global average pooling* final
- Maior variante: ~420M parâmetros

Variantes ViT

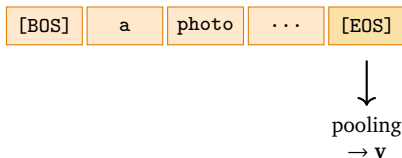
- ViT-B/32, ViT-B/16, ViT-L/14
- Patches de 14×14 , 16×16 ou 32×32
- ViT-L/14 atinge melhor desempenho final
- ~304M parâmetros (ViT-L/14)

Saída do *backbone*: vetor de dimensão $d \in \{512, 768, 1024\}$, depois projetado linearmente para $d_e = 512$ (ou 768 para ViT-L/14).

Encoder de Texto: Transformer Compacto

Detalhes do *transformer* de texto:

- 12 camadas, dimensão 512, 8 cabeças de atenção
- Tokenização *Byte-Pair Encoding* (BPE), 49408 tokens
- Sequência máxima de 77 tokens
- Tokens especiais [BOS] e [EOS]
- Pooling: **representação do token** [EOS] na última camada
- ~63M parâmetros



Cabeças de Projeção: Mesmo Espaço para Texto e Imagem

Cada *encoder* produz um vetor em sua dimensão nativa:

$$\mathbf{h}_i^{\text{img}} \in \mathbb{R}^{d_v}, \quad \mathbf{h}_t^{\text{txt}} \in \mathbb{R}^{d_t}$$

Projetamos linearmente para a **mesma dimensão** d_e e normalizamos:

$$\mathbf{u}_i = \frac{W_i \mathbf{h}_i^{\text{img}}}{\|W_i \mathbf{h}_i^{\text{img}}\|}, \quad \mathbf{v}_t = \frac{W_t \mathbf{h}_t^{\text{txt}}}{\|W_t \mathbf{h}_t^{\text{txt}}\|}$$

com $W_i \in \mathbb{R}^{d_e \times d_v}$ e $W_t \in \mathbb{R}^{d_e \times d_t}$.

Por que a normalização?

A perda usa cosseno; sem L2 a temperatura τ teria que absorver magnitudes diferentes entre os dois ramos.

Pseudocódigo: *Forward + Loss*

- Entrada: batch de N imagens e N textos correspondentes
- Saída: **escalar** $\mathcal{L}_{\text{CLIP}}$
- As duas *cross-entropies* compartilham a mesma matriz S

Require: batch $\{(\text{img}_i, \text{txt}_i)\}_{i=1}^N, \tau$

- 1: $\mathbf{H}^{\text{img}} \leftarrow \text{ImageEncoder}(\text{img})$
- 2: $\mathbf{H}^{\text{txt}} \leftarrow \text{TextEncoder}(\text{txt})$
- 3: $U \leftarrow \text{normalize}(W_i \mathbf{H}^{\text{img}})$
- 4: $V \leftarrow \text{normalize}(W_t \mathbf{H}^{\text{txt}})$
- 5: $S \leftarrow UV^\top / \tau$
- 6: $\mathbf{y} \leftarrow [0, 1, \dots, N-1]$
- 7: $\mathcal{L}_1 \leftarrow \text{CE}(S, \mathbf{y})$
- 8: $\mathcal{L}_2 \leftarrow \text{CE}(S^\top, \mathbf{y})$
- 9: **return** $(\mathcal{L}_1 + \mathcal{L}_2)/2$

Custos Computacionais e Memória do *Batch*

Variante	Params (enc. imagem)	FLOPs/imagem	Memória de S ($N=32k$)	Hardware (paper)
ResNet-50	~38M	4.1 GFLOPs	~4 GB (fp32)	—
ResNet-50x64	~420M	165 GFLOPs	~4 GB	592 V100 / 18 dias
ViT-B/32	~88M	4.4 GFLOPs	~4 GB	—
ViT-L/14	~304M	81 GFLOPs	~4 GB	256 V100 / 12 dias

- Memória de S cresce **quadraticamente** com N
- Truques: *gradient checkpointing*, *mixed precision*, *shard* de embeddings
- **Compute dominado pelos encoders:** $\sim N \cdot \text{GFLOPs/imagem}$ contra $\sim N^2 d_e$ FLOPs do produto UV^T que forma S (ordens de magnitude de diferença)
- **Memória dominada pelas ativações dos encoders** quando S é distribuído via *sharded embeddings*; sem *shard*, S (~ 4 GB) seria competitivo

Visão Geral

1. Motivação e Contexto
2. Aprendizado Contrastivo
3. Arquitetura do CLIP
- 4. Dataset e Treinamento**
5. Avaliação e Uso
6. Limitações e Sucessores

WIT-400M: O Combustível do CLIP

- **WIT** (*WebImageText*): ~400M pares (imagem, texto) coletados da internet pela OpenAI
- Pipeline: começa com ~500 000 *queries* de busca; equilibra resultados por consulta
- Filtragem leve (deduplicação, descarta textos muito curtos), **sem rotulação humana**
- Conjunto **fechado** (não público); réplicas abertas posteriores: LAION-400M, LAION-5B

Filosofia anti-curadoria

Ruído residual é compensado por escala. “Mais dados, mesmo barulhentos” supera “menos dados, perfeitamente rotulados” nesse regime.

Datasets de Pré-treino: Ordem de Grandeza

Dataset	Modalidade	Pares/imagens	Rotulagem	Acesso
ImageNet-1k	imagem (1k classes)	1.28M	humana	público
ImageNet-21k	imagem (21k classes)	14M	humana	público
JFT-300M (Google)	imagem	300M	semi-aut.	fechado
WIT-400M (OpenAI)	imagem + texto	400M	web	fechado
LAION-400M ³	imagem + texto	400M	web filtr.	aberto
LAION-5B	imagem + texto	5.85B	web filtr.	aberto

- LAION usa o próprio CLIP para filtrar pares com baixa similaridade
- Réplicas abertas permitem reproduzir CLIP em escala (OpenCLIP)

³Christoph Schuhmann et al. "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models". Em: [NeurIPS](#), vol. 35, 2022, pp. 25278–25294.

Receita de Treinamento

Hiperparâmetros

- Otimizador: **AdamW**, $\beta = (0.9, 0.98)$
- *Weight decay* desacoplado, 0.2
- Taxa de aprendizado com *cosine schedule*
- Batch size: **32 768**
- *Mixed precision* (fp16) com *loss scaling*
- τ **aprendido**, *clipped* em $\log 100$

Computação real (paper)

- ResNet-50x64: **18 dias** em **592 V100**
- ViT-L/14: **12 dias** em **256 V100**
- 32 épocas sobre 400M pares

Hiperparâmetros encontrados via *mix* de *grid*, *random* e *manual tuning* em ResNet-50 na primeira época.

Truques para Escalar o *Batch*

- *Sharded embeddings*: distribuir U e V entre GPUs; cada GPU calcula sua fatia de S
- *Gradient checkpointing* nos *encoders* pesados (ViT-L)
- *Mixed precision* reduz memória dos forward passes pela metade
- *Local contrastive*: cada GPU vê o batch global de V via *all-gather*, calcula sua linha de S e devolve gradientes

Pergunta comum

Precisamos minerar negativos difíceis (*hard negative mining*)?

Não no caso de CLIP. O batch de 32 768 já fornece um espectro suficiente de negativos a cada passo.

Visão Geral

1. Motivação e Contexto
2. Aprendizado Contrastivo
3. Arquitetura do CLIP
4. Dataset e Treinamento
- 5. Avaliação e Uso**
6. Limitações e Sucessores

Linear Probe Evaluation

Para medir a qualidade dos *embeddings* aprendidos, congelamos o *encoder* e treinamos apenas um classificador linear:

1. Para cada imagem $\mathbf{x}$, computar $\mathbf{u} = f_{\text{CLIP}}(\mathbf{x})$
2. Treinar regressão logística sobre $(\mathbf{u}, y)$ no *dataset* alvo
3. Reportar *accuracy* no conjunto de teste

Por que esse protocolo?

Mede a separabilidade linear das classes no espaço de representação, isolando a qualidade dos *features* de qualquer ajuste fino do *backbone*.

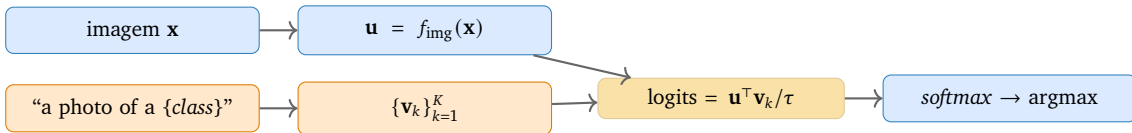
Comparação direta com SimCLRv2, BYOL, MoCo-v3 e ResNet supervisionada na ImageNet.

Resultados *Headline* do Paper Original

Método	ImageNet <i>zero-shot</i>	ImageNet <i>linear probe</i>	ObjectNet <i>zero-shot</i>
ResNet-50 supervisionada	—	76.1%	32.6%
SimCLRv2 (linear)	—	79.8%	—
CLIP ViT-L/14	75.5%	83.9%	72.3%
CLIP ViT-L/14@336	76.2%	85.4%	72.3%

- Zero-shot CLIP **empata** com ResNet-50 supervisionada em ImageNet
- Em ObjectNet (distribuição shiftada) o ganho é dramático: 32.6% → 72.3%
- *Robustness*: CLIP cai muito menos sob mudança de domínio

Zero-Shot Classification: o Pipeline



Os K *embeddings* de texto são **calculados uma vez** e reutilizados para toda imagem que chega.

Funciona porque o treino contrastivo alinhou texto e imagem no **mesmo espaço**: uma classe nova, descrita só por texto, cai perto das imagens correspondentes, mesmo sem nenhum exemplo rotulado.

Zero-Shot na Prática: Imagem de Entrada e *Prompts*



Imagem de entrada $\mathbf{x}$

Três classes candidatas, descritas por *prompts* de texto:

- “a photo of a **gato**”
- “a photo of a **cachorro**”
- “a photo of a **carro**”

Em produção, esse conjunto pode ser trocado a qualquer momento, sem re-treino.

No próximo slide

Codificamos a imagem em $\mathbf{u}$, cada *prompt* em $\mathbf{v}_k$, e aplicamos *softmax* sobre $s_k = \mathbf{u}^\top \mathbf{v}_k$.

Exemplo Trabalhado III: *Zero-Shot* com 3 Classes

Embedding da imagem do slide anterior: $\mathbf{u} = (0.6, 0.8, 0)^\top$. *Embeddings* dos *prompts*:

$$\mathbf{v}_{\text{gato}} = (0, 1.0, 0)^\top, \mathbf{v}_{\text{cachorro}} = (0.8, 0.6, 0)^\top, \mathbf{v}_{\text{carro}} = (0, 0, 1.0)^\top$$

Similaridades: $s_{\text{gato}} = 0.80$, $s_{\text{cachorro}} = 0.96$, $s_{\text{carro}} = 0.00$

Com $\tau = 0.1$, logits = (8.0, 9.6, 0). Subtraindo o máximo:

- $e^{-1.6} \approx 0.202$, $e^0 = 1$, $e^{-9.6} \approx 6.77 \times 10^{-5}$
- soma ≈ 1.202
- $p_{\text{gato}} \approx 0.168$, $p_{\text{cachorro}} \approx 0.832$, $p_{\text{carro}} \approx 0.00006$

Predição

$\hat{y} = \operatorname{argmax}_k p_k = \text{cachorro}$. O classificador é montado **em tempo de inferência** a partir dos *prompts*; basta trocar as descrições para mudar o conjunto de classes.

Prompt Engineering e Ensemble de Templates

A escolha do *template* muda a *accuracy* em vários pontos percentuais:

- "{class}" (apenas o nome): *baseline*
- "a photo of a {class}": +1 a +2 pp em ImageNet
- Templates específicos por dataset: "a satellite photo of a {class}", "a sketch of a {class}"

Ensemble de prompts

O paper original usa **80 templates** por classe e calcula a média dos *embeddings* de texto antes da L2:

$$\mathbf{v}_k = \text{normalize} \left(\frac{1}{T} \sum_{t=1}^T f_{\text{txt}}(\text{prompt}_t(k)) \right)$$

Ganho médio em ImageNet: +3.5 pp.

Adaptação Eficiente: Linear Head, Adapter, LoRA

Cabeçalho linear

Congelar CLIP, treinar uma camada $W \in \mathbb{R}^{K \times d_e}$ sobre os *embeddings*.

Linear probe clássico, baixíssimo custo.

CLIP-Adapter

Pequeno MLP residual sobre $\mathbf{u}$, com mistura ponderada:

$$\mathbf{u}' = \alpha \text{MLP}(\mathbf{u}) + (1 - \alpha) \mathbf{u}.$$

LoRA sobre os *encoders*

Adicionar matrizes de baixa ordem $W + \Delta = W + BA$ nos *transformers* de imagem e texto.

Apenas 0.1–1% dos parâmetros treinados.

Quando NÃO fazer fine-tuning

Se a tarefa cabe em *zero-shot* robusto, o *fine-tuning* piora a generalização *out-of-distribution*.

Semantic Retrieval: Buscar Imagens com Texto

1. **Indexação** (offline): para cada imagem $\mathbf{x}_n$ no acervo, salvar $\mathbf{u}_n = f_{\text{img}}(\mathbf{x}_n)$
2. Construir índice *Approximate Nearest Neighbor* (FAISS, HNSW, ScaNN) sobre $\{\mathbf{u}_n\}$ usando produto interno
3. **Consulta** (online): dado um texto q , computar $\mathbf{v}_q = f_{\text{txt}}(q)$
4. Recuperar *top-k* pelo cosseno $\mathbf{v}_q^\top \mathbf{u}_n$

Custo amortizado

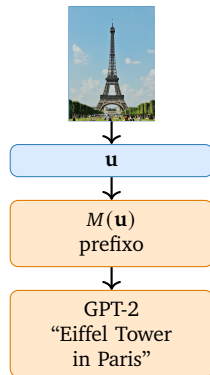
O *encoder* de imagem roda **uma vez** por imagem do acervo; consultas custam apenas um forward de texto + *look-up* no índice.

Inverso: dada uma imagem, recuperar legendas similares no acervo de textos.

Image Captioning com CLIP⁴

ClipCap: usar o *embedding* de CLIP como **prefixo** para um modelo de linguagem (GPT-2) gerar a legenda.

1. Imagem $\rightarrow \mathbf{u} = f_{\text{img}}(\mathbf{x})$
2. Mapeamento $M : \mathbb{R}^{d_e} \rightarrow \mathbb{R}^{L \times d_{\text{LM}}}$ (*prefix*)
3. GPT-2 gera tokens condicionado nesse prefixo
4. Treinar M (e opcionalmente fazer fine-tuning do GPT-2)



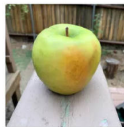
⁴Ron Mokady, Amir Hertz e Amit H. Bermano. "ClipCap: CLIP Prefix for Image Captioning". Em: [arXiv preprint arXiv:2111.09734](https://arxiv.org/abs/2111.09734) (2021).

Visão Geral

1. Motivação e Contexto
2. Aprendizado Contrastivo
3. Arquitetura do CLIP
4. Dataset e Treinamento
5. Avaliação e Uso
- 6. Limitações e Sucessores**

Falhas Conhecidas do CLIP

- **Texto na imagem:** CLIP “lê” palavras pintadas e envia a classificação (uma maçã com etiqueta "iPod" é classificada como iPod)
- **Generalização para MNIST:** 88% *accuracy* (pior que regressão logística simples), dígitos manuscritos são raros em pares web
- **Contagem de objetos:** “três cachorros” vs. “cinco cachorros” é difícil
- **Granularidade fina:** distinguir espécies próximas exige supervisão dedicada
- **Tarefas espaciais:** localização precisa, profundidade, geometria



Granny Smith	85.6%
iPod	0.4%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.1%



Granny Smith	0.1%
iPod	99.7%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.0%

Vieses Herdados da Web

O paper original dedica uma seção a auditoria de vieses:

- Classificação errônea sistemática de pessoas negras como categorias não-humanas em FairFace
- Associações de gênero em prompts ocupacionais (“CEO” enviesada para masculino)
- Sensibilidade a vocabulário ofensivo presente nos pares web

Implicação prática

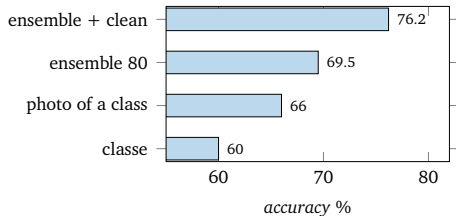
Antes de usar CLIP em produto: auditar para o seu domínio, considerar *prompt* sanitizado, restringir vocabulário de classes, registrar limites do modelo.

Sensibilidade a *Prompt*

- Trocar "{class}" por "a photo of a {class}" pode mudar *accuracy* em 5–10 pp
- Templates errados podem inverter ranqueamentos
- Sem *ground truth*, calibrar prompt requer um conjunto de validação

Em produção, recomenda-se:

- Ensemble de 5–80 templates
- Avaliar *accuracy* em conjunto de validação rotulado



Sucessores: ALIGN, OpenCLIP, SigLIP

ALIGN (Jia et al., 2021)

- Google, contemporâneo do CLIP
- 1.8B pares (alt-text bruto, ainda mais ruído)
- Mostra que limpeza importa menos que escala

OpenCLIP (Cherti et al., 2023)

- Reprodução aberta sobre LAION
- Estuda *scaling laws* multimodais
- Pesos públicos para múltiplas variantes

SigLIP (2023)

Substitui *softmax* por **sigmoide** pareada:

$$\ell_{ij} = \log\left(1 + e^{-y_{ij} \mathbf{u}_i^\top \mathbf{v}_j / \tau}\right)$$

- Não exige *batch* grande
- Cada par é independente
- Ganho prático em *batches* pequenos

EVA-CLIP, MetaCLIP, DFN

Variações em receita de dados, escala e arquitetura.

LAION-5B e a Democratização do Pré-treino Multimodal

- **5.85B** pares (imagem, texto), abertos, multilíngues
- Filtragem usa CLIP: descarta pares com baixa similaridade $\Rightarrow$ recursão útil
- Possibilita treinar variantes de CLIP, Stable Diffusion, vários modelos abertos

Implicações éticas

- Conteúdo problemático (NSFW, vieses, material sensível) presente
- Direitos autorais: imagens coletadas sem consentimento explícito
- Auditoria: `laion.ai` disponibiliza ferramentas de inspeção

Leituras Recomendadas I

- Paper original do CLIP: Alec Radford et al. “Learning Transferable Visual Models from Natural Language Supervision”. Em: [ICML](#). 2021, pp. 8748–8763
- InfoNCE e *contrastive predictive coding*: Aaron van den Oord, Yazhe Li e Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding”. Em: [arXiv preprint arXiv:1807.03748](#) (2018)
- ALIGN (contemporâneo, Google): Chao Jia et al. “Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision”. Em: [ICML](#). 2021, pp. 4904–4916
- OpenCLIP e *scaling laws*: Mehdi Cherti et al. “Reproducible Scaling Laws for Contrastive Language-Image Learning”. Em: [CVPR](#). 2023, pp. 2818–2829
- LAION-5B (dataset aberto): Christoph Schuhmann et al. “LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models”. Em: [NeurIPS](#). vol. 35. 2022, pp. 25278–25294

Leituras Recomendadas II

- ClipCap (*captioning* via prefix): Ron Mokady, Amir Hertz e Amit H. Bermano. “ClipCap: CLIP Prefix for Image Captioning”. Em: [arXiv preprint arXiv:2111.09734](#) (2021)

Referências I

- [1] Mehdi Cherti et al. “Reproducible Scaling Laws for Contrastive Language-Image Learning”. Em: [CVPR](#). 2023, pp. 2818–2829.
- [2] Chao Jia et al. “Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision”. Em: [ICML](#). 2021, pp. 4904–4916.
- [3] Ron Mokady, Amir Hertz e Amit H. Bermano. “ClipCap: CLIP Prefix for Image Captioning”. Em: [arXiv preprint arXiv:2111.09734](#) (2021).
- [4] Aaron van den Oord, Yazhe Li e Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding”. Em: [arXiv preprint arXiv:1807.03748](#) (2018).
- [5] Alec Radford et al. “Learning Transferable Visual Models from Natural Language Supervision”. Em: [ICML](#). 2021, pp. 8748–8763.
- [6] Christoph Schuhmann et al. “LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models”. Em: [NeurIPS](#). Vol. 35. 2022, pp. 25278–25294.