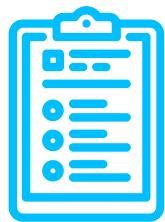


Sesión 2.2

Training and Representations

Tricks, Generalización

1.

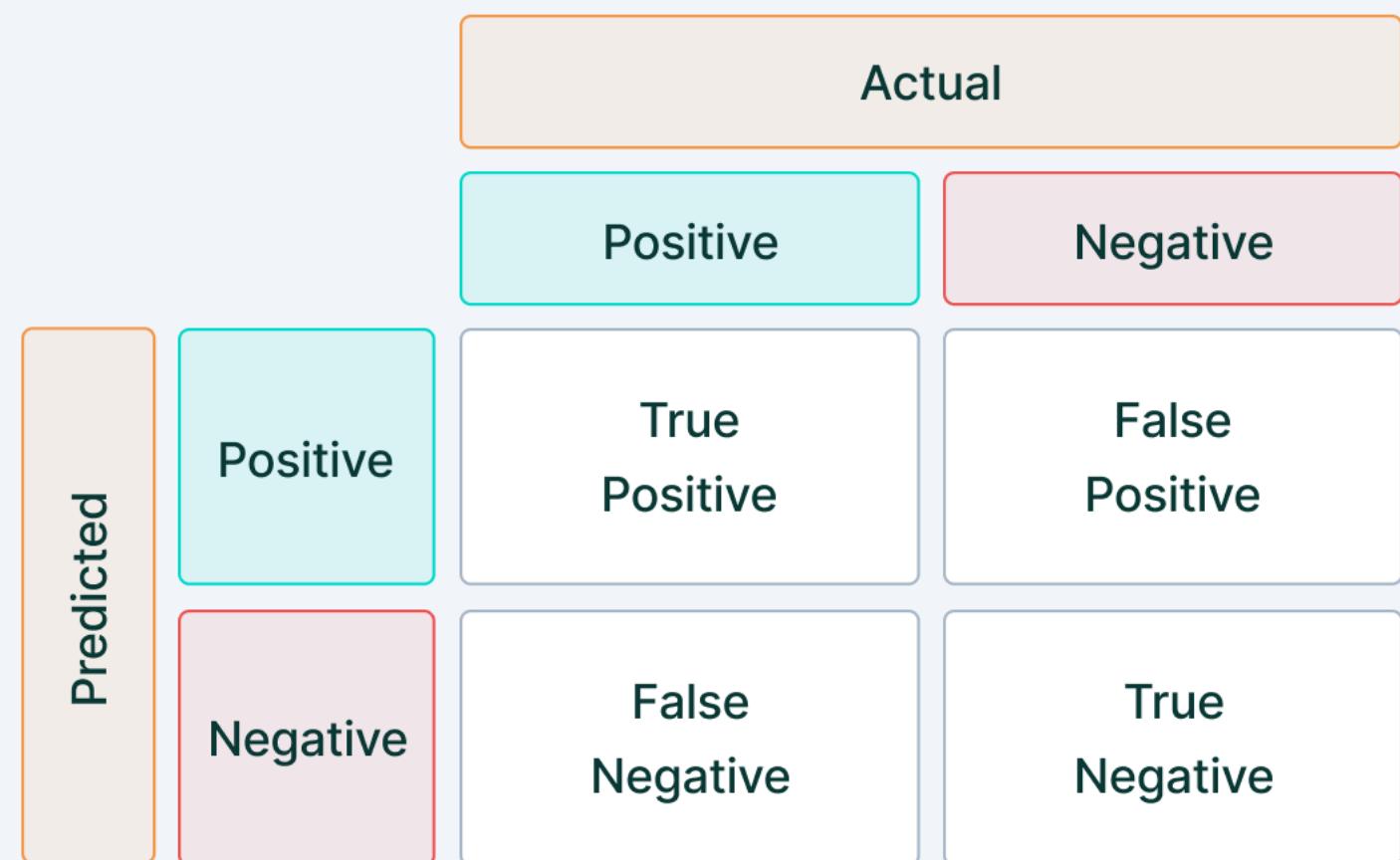


Metrics

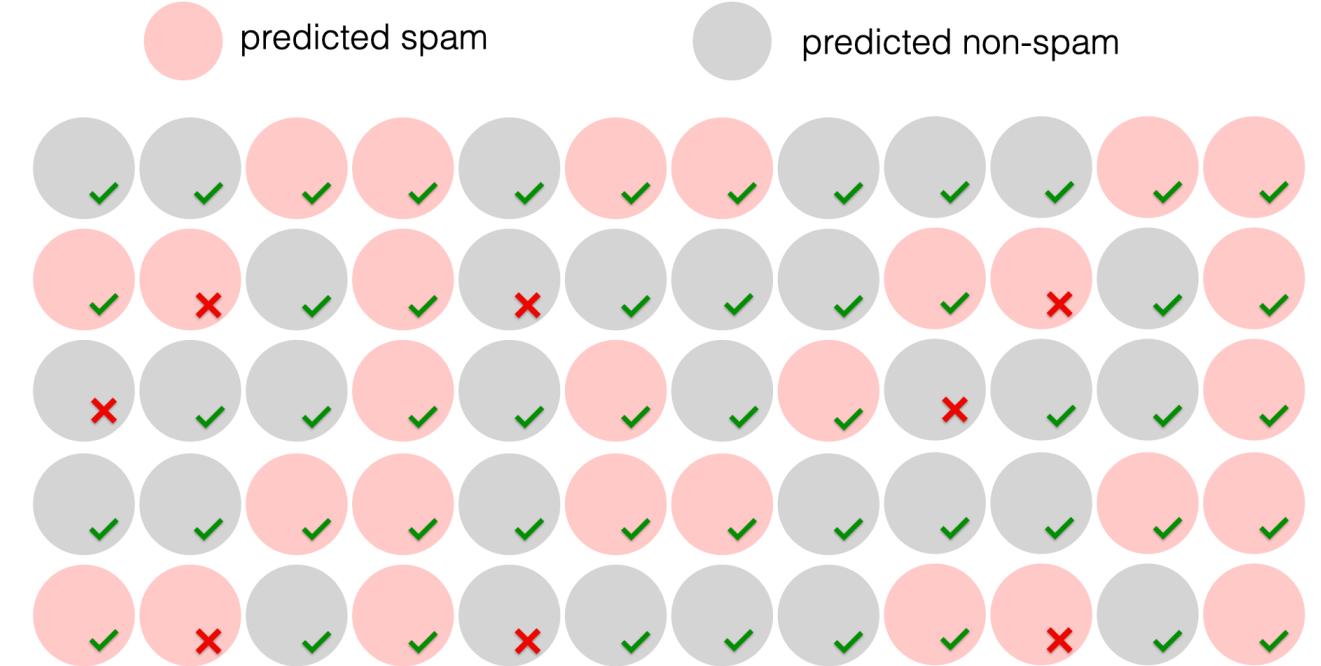
TRANSFORMATEC



Métricas



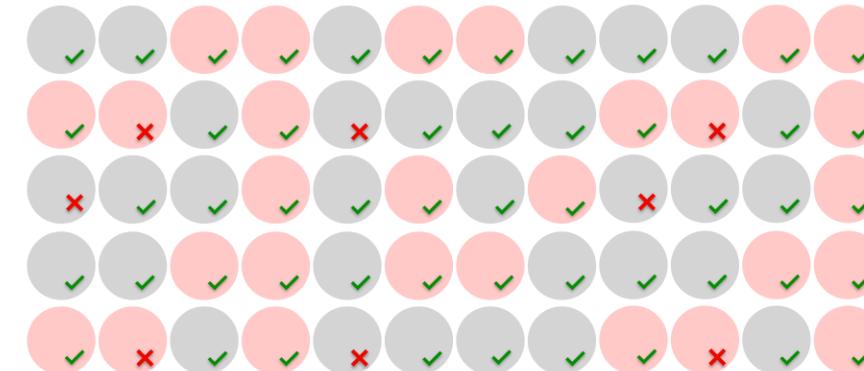
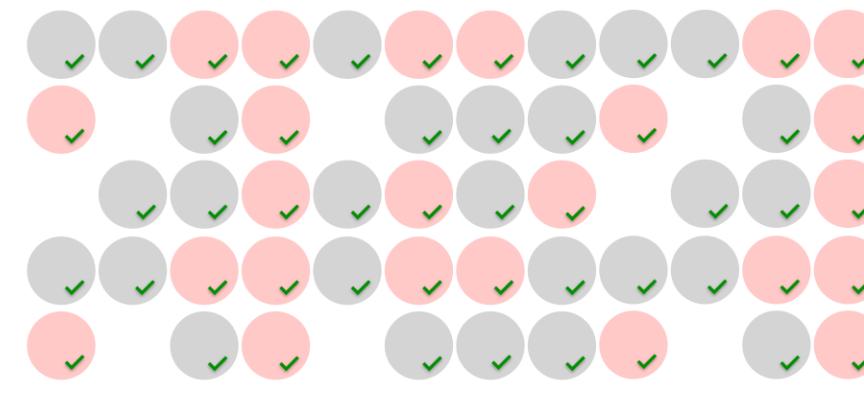
Accu*racy*



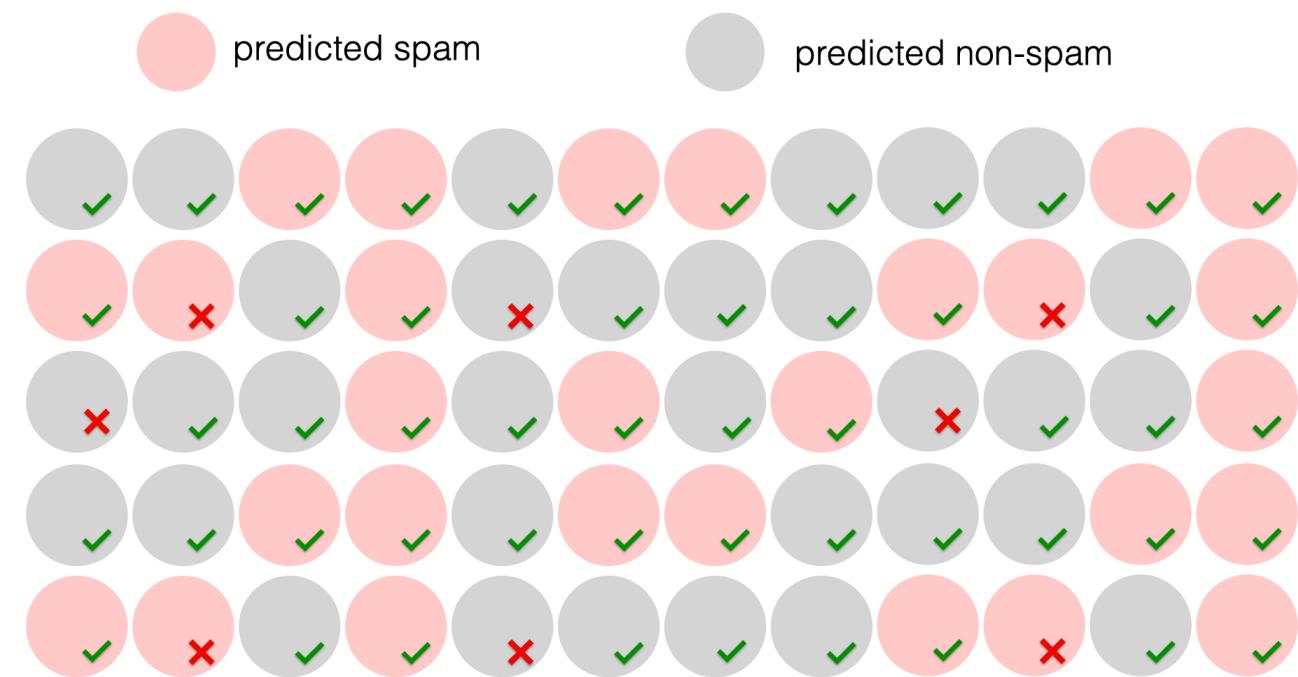
correct predictions

Accuracy =

all predictions



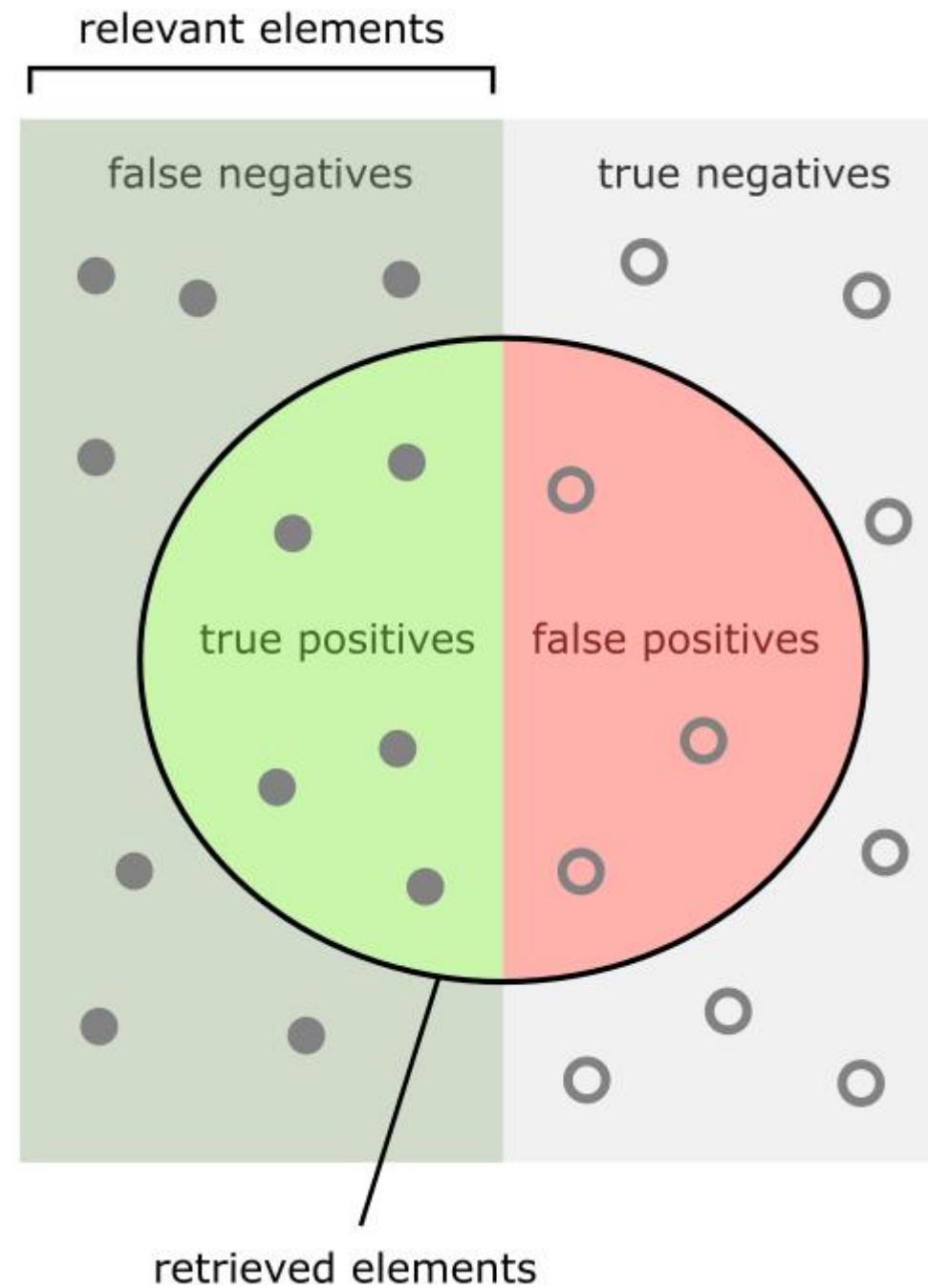
Accu*racy*



$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$



Sensitivity vs Specificity



¿Cuántos elementos relevantes se seleccionan?

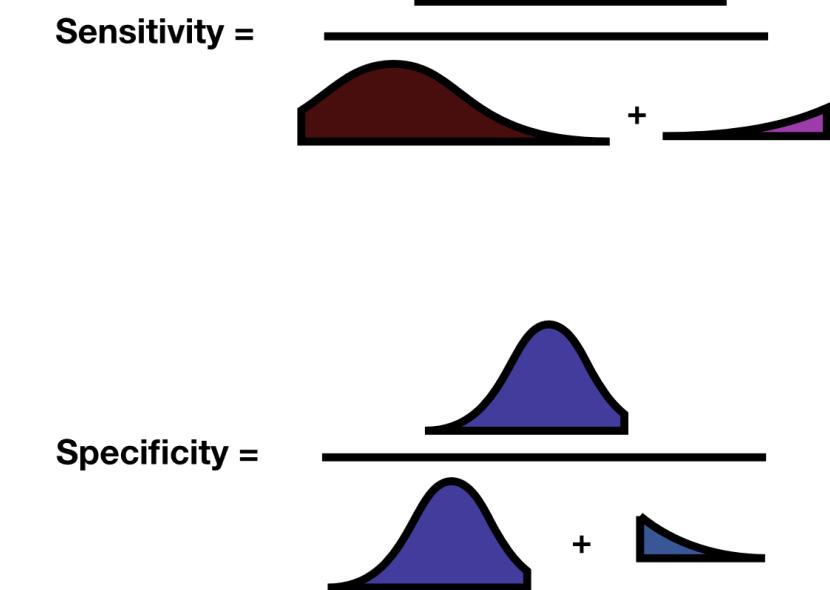
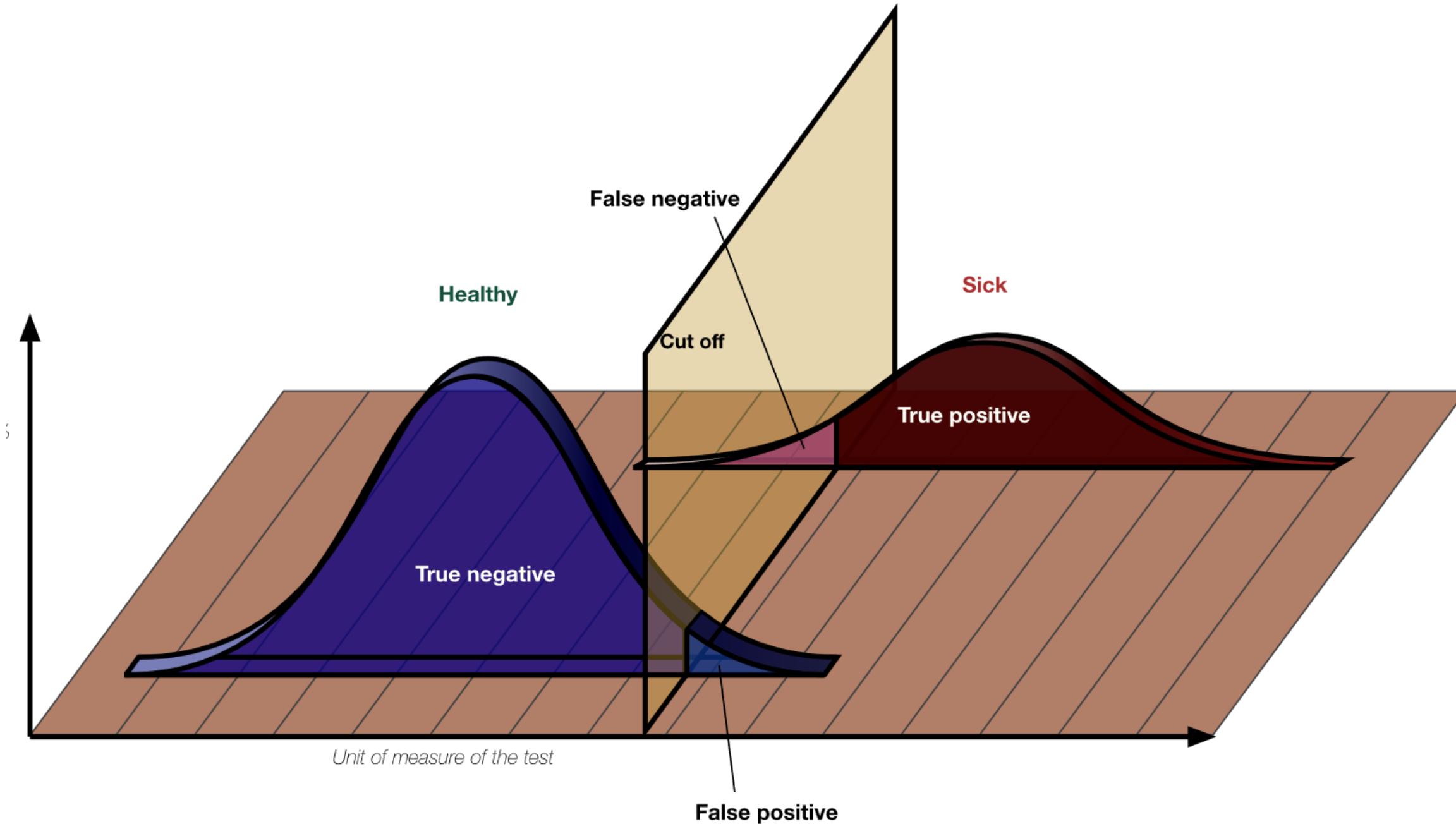
$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

¿Cuántos elementos negativos seleccionados son realmente negativos?

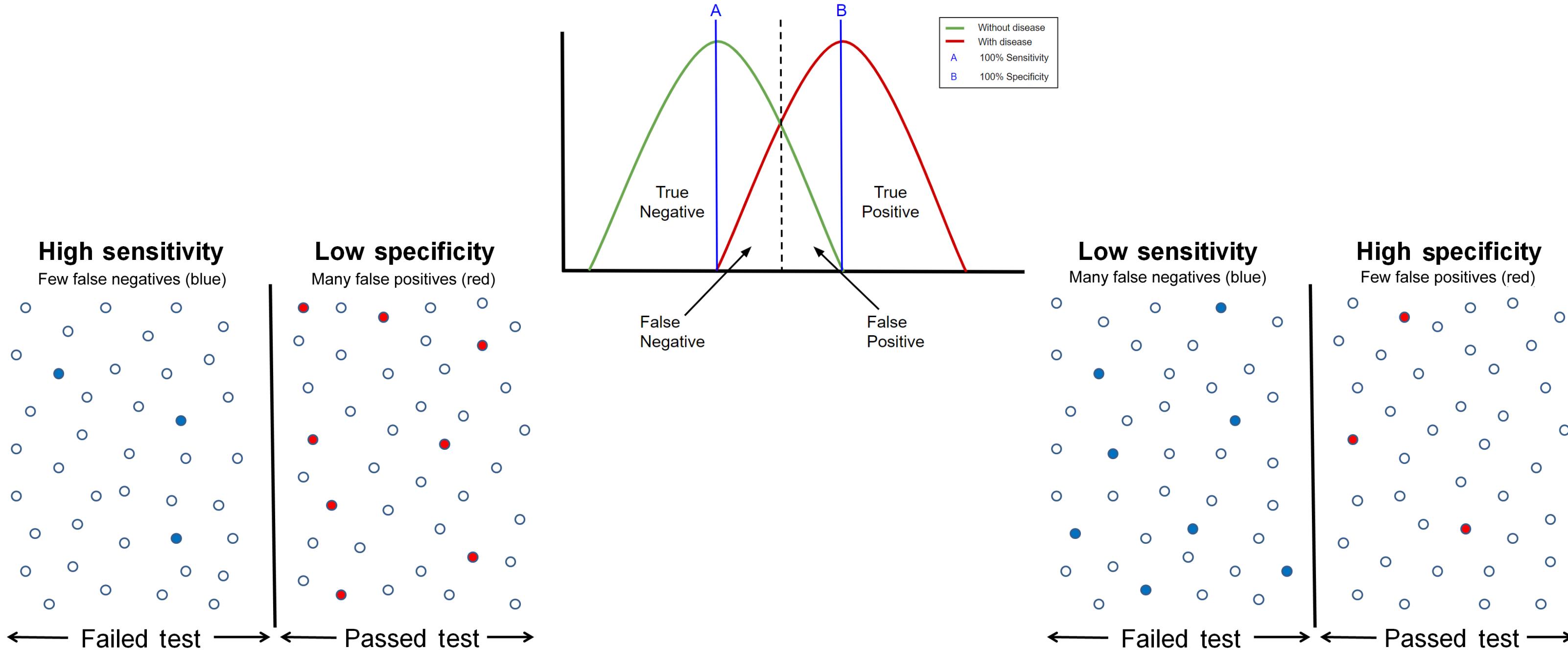
$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$



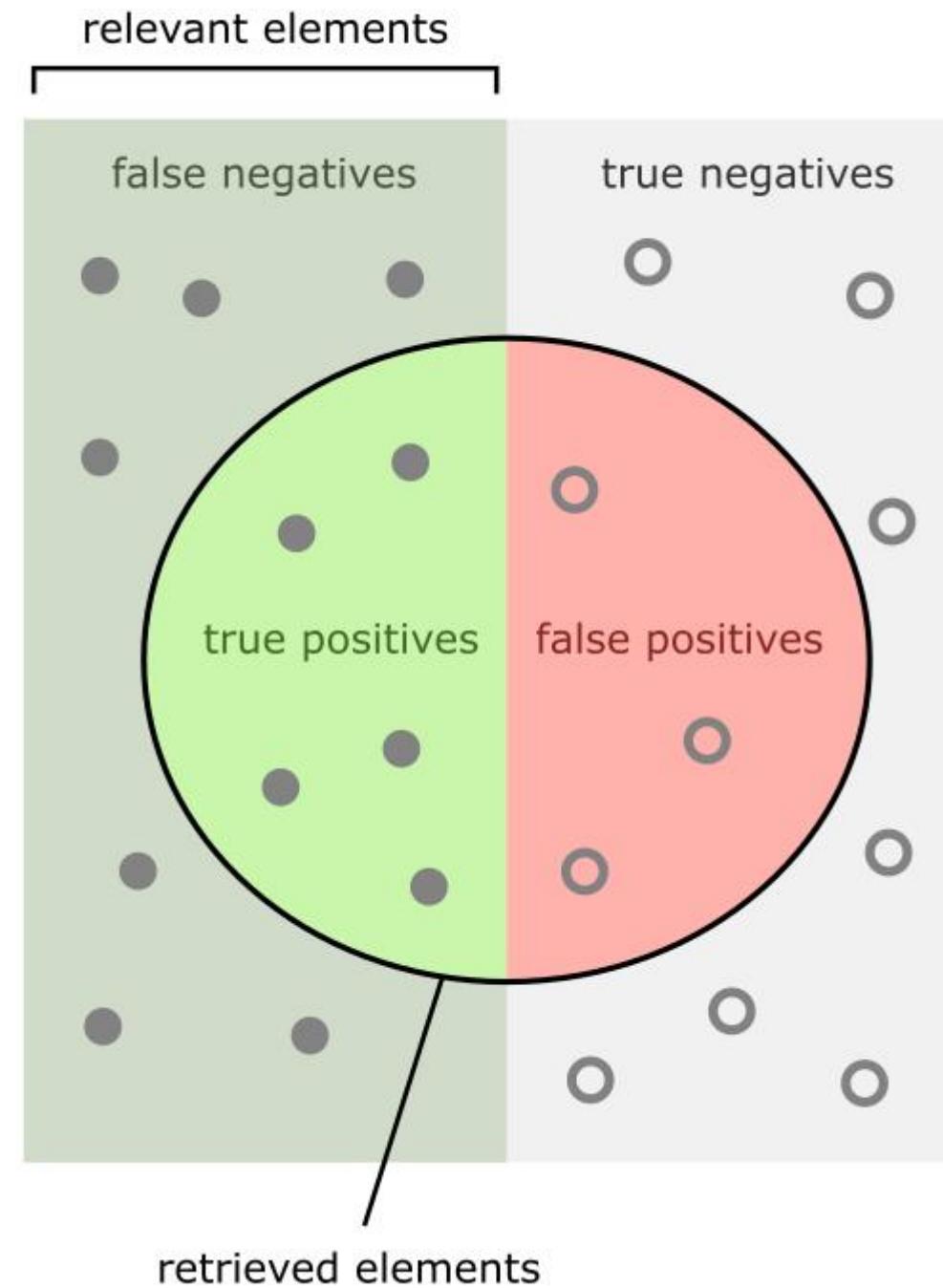
Sensitivity vs Specificity



Sensitivity vs Specificity



Métricas



¿Cuántos elementos recuperados son relevantes?

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

¿Cuántos elementos relevantes se recuperan?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$



F1-score

El F1-score es una métrica de evaluación utilizada en problemas de clasificación para encontrar un balance entre la precision y el recall.

$$F1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$



F1-score

F1-micro

- Calcula globalmente las verdaderas positivas (TP), falsas positivas (FP) y falsas negativas (FN), y luego aplica la fórmula del F1-score.
- Pondera por la frecuencia de cada clase y es útil cuando hay un desequilibrio de clases.

$$F1_{micro} = \frac{2 \cdot \sum TP}{2 \cdot \sum TP + \sum FP + \sum FN}$$

Se emplea cuando se necesita evaluar el rendimiento global, especialmente en datasets desbalanceados.



F1-score

F1-macro

- Calcula el F1-score para cada clase de manera individual y luego obtiene el promedio.
- Trata todas las clases por igual, sin importar su frecuencia.
- Útil si quieres evaluar el rendimiento en todas las clases de manera equitativa.

$$F1_{macro} = \frac{1}{n} \sum_{i=1}^n \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

Se emplea cuando quieras dar la misma importancia a cada clase, sin importar su frecuencia.



F1-score

F1-weighted

- Similar al F1-macro, pero pondera el F1-score de cada clase por el número de muestras de esa clase.
- Proporciona una métrica más representativa si el dataset está desbalanceado.

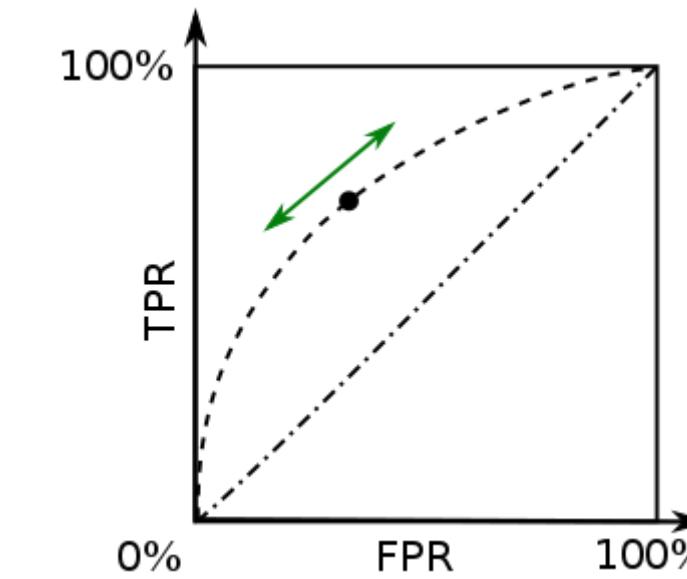
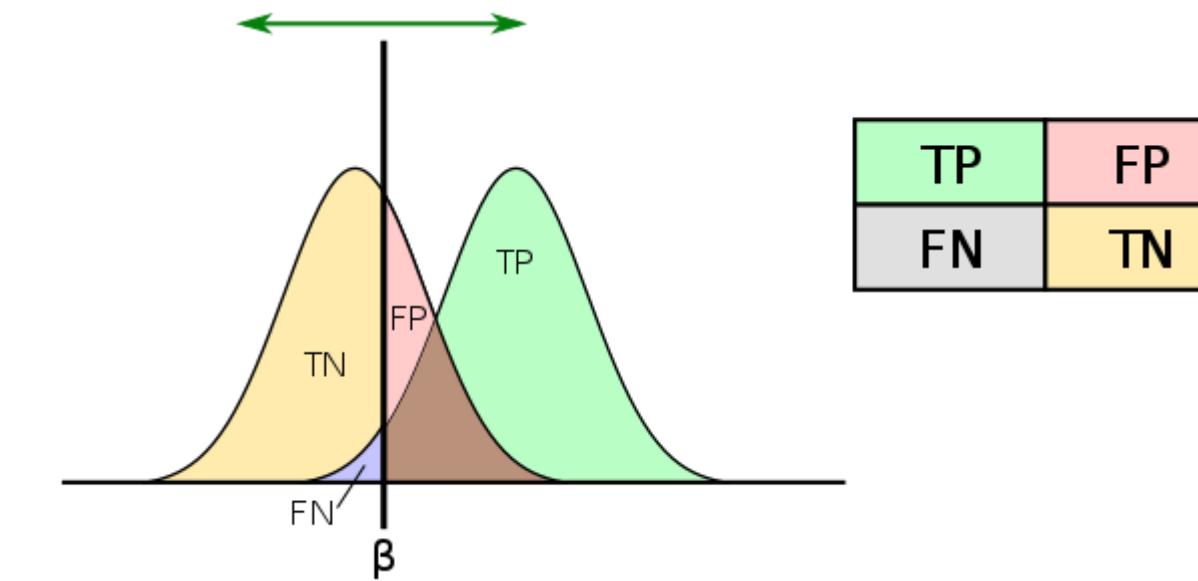
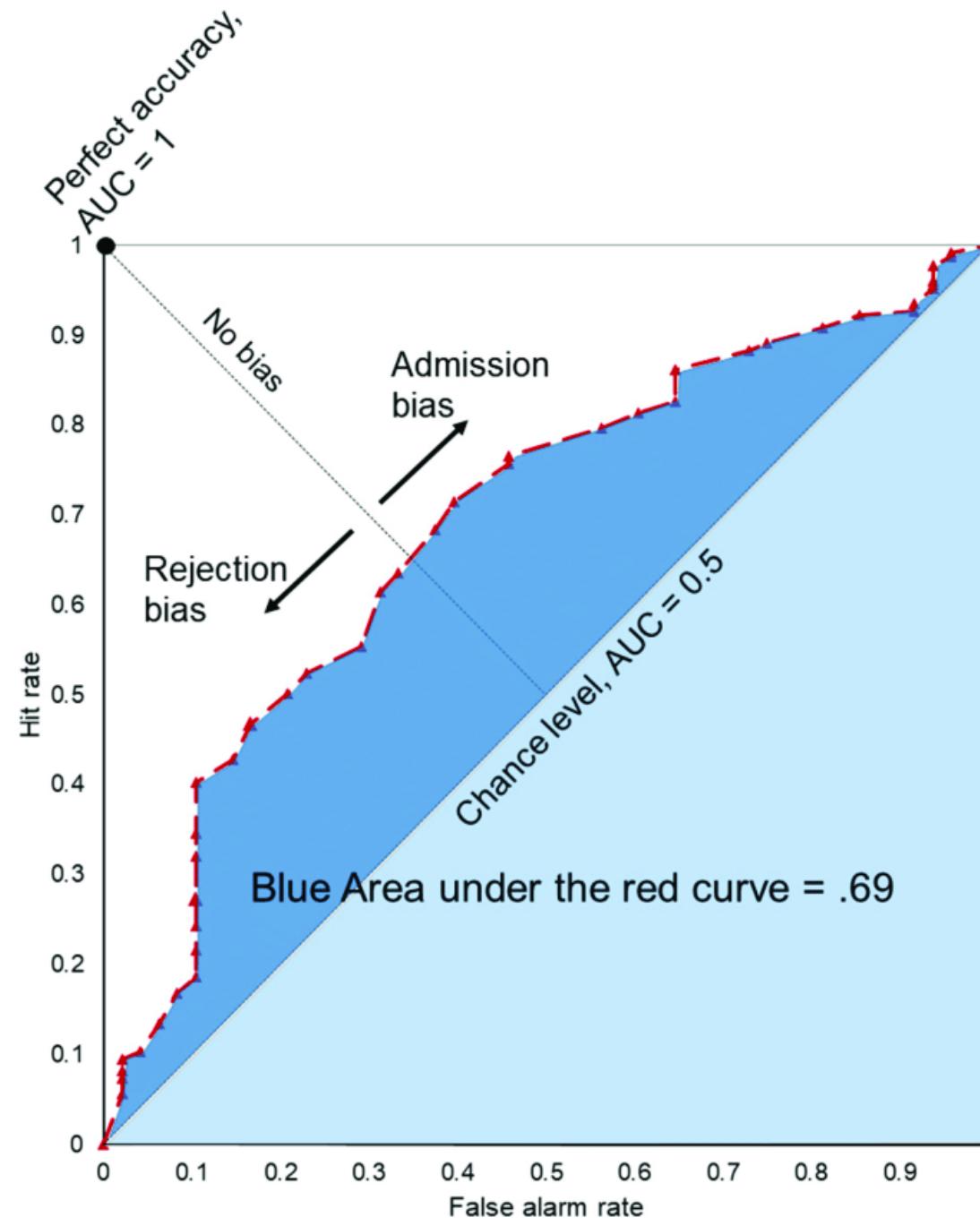
$$F1_{weighted} = \sum_{i=1}^n w_i \cdot F1_i$$

donde w_i es la proporción de la clase i .

Se emplea cuando el desbalance de clases es significativo, pero aún quieras reflejar el tamaño de cada clase en la evaluación.



Receiver Operating Characteristic curve (ROC)



True Positive Rate (recall)

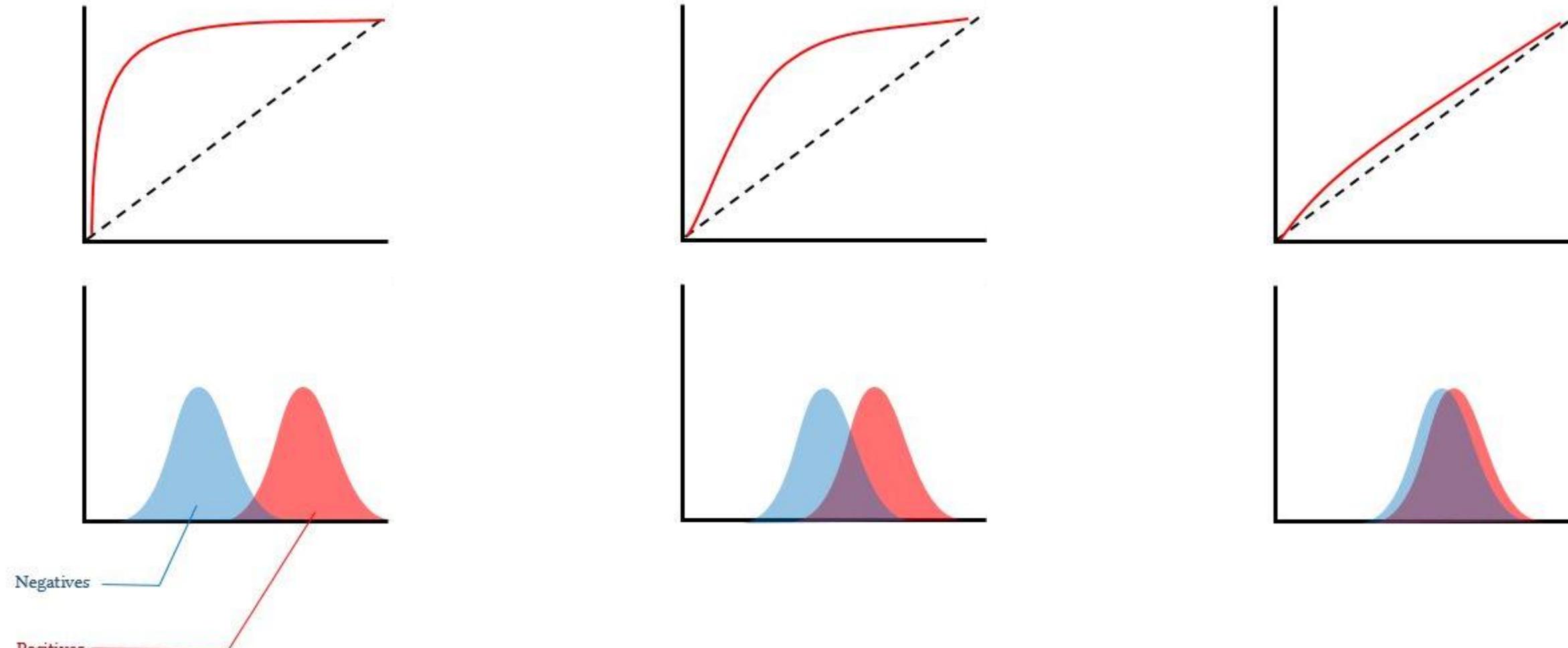
$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate

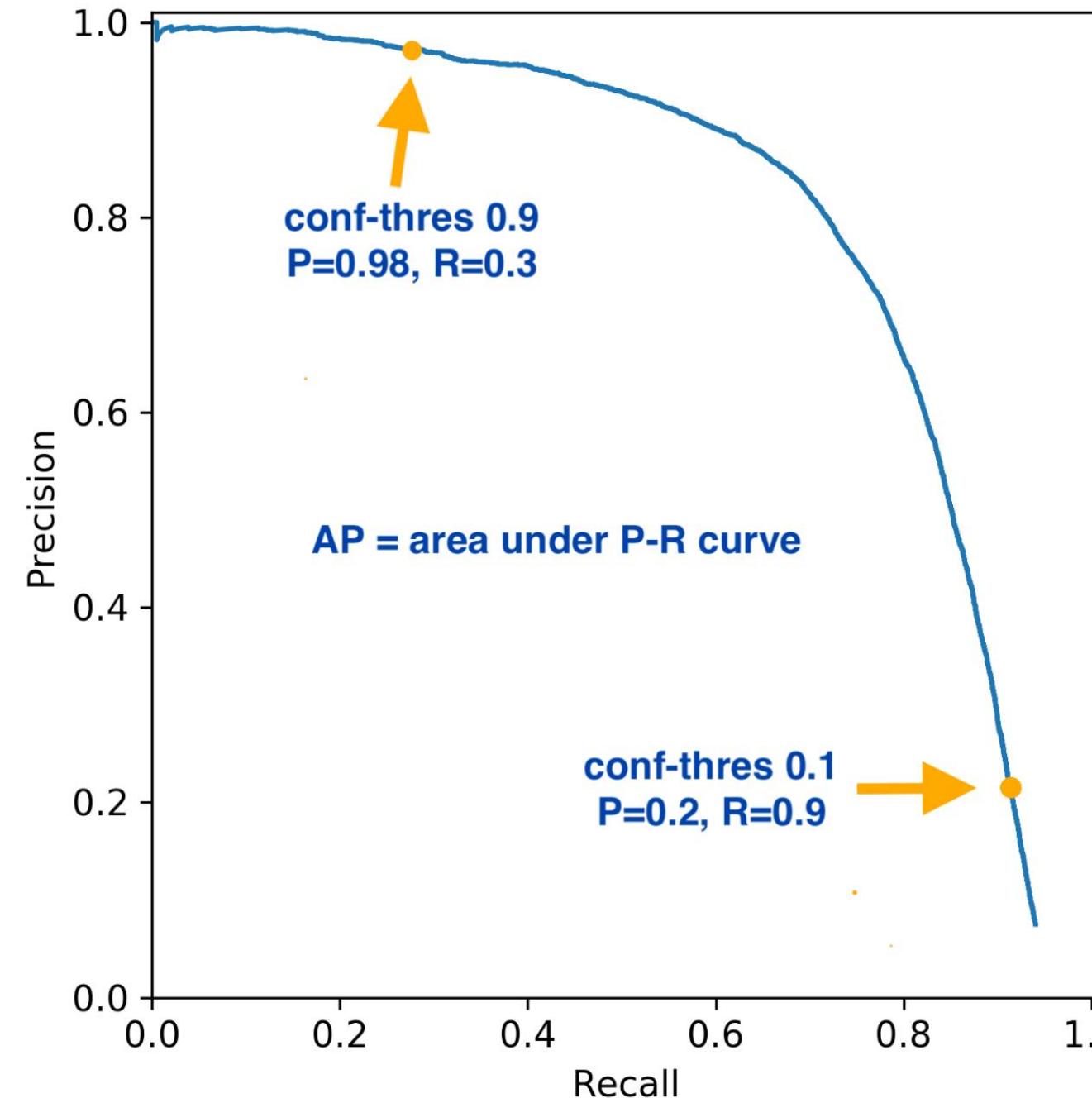
$$FPR = \frac{FP}{FP + TN}$$



Receiver Operating Characteristic curve (ROC)



Precision-*Recall Curve*



$$AP = \sum_{n=1}^N (R_n - R_{n-1})P_k$$

donde P_n y R_n son precision y recall en el n-ésimo threshold.
Al par (P_n, R_n) es conocido como operating point.



Métricas

Receiver Operating Characteristic curve

Muestra la habilidad general de clasificación.

Cuando usar

- Cuando las clases son balanceadas (o aproximadamente balanceadas).
- Cuando se desea evaluar la discriminación global del modelo.

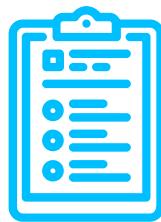
Precision-Recall Curve

Muestra el rendimiento del modelo sobre la clase positiva.

- Cuando los datos son altamente desbalanceados.
- Cuando el interés principal es reducir los falsos positivos.



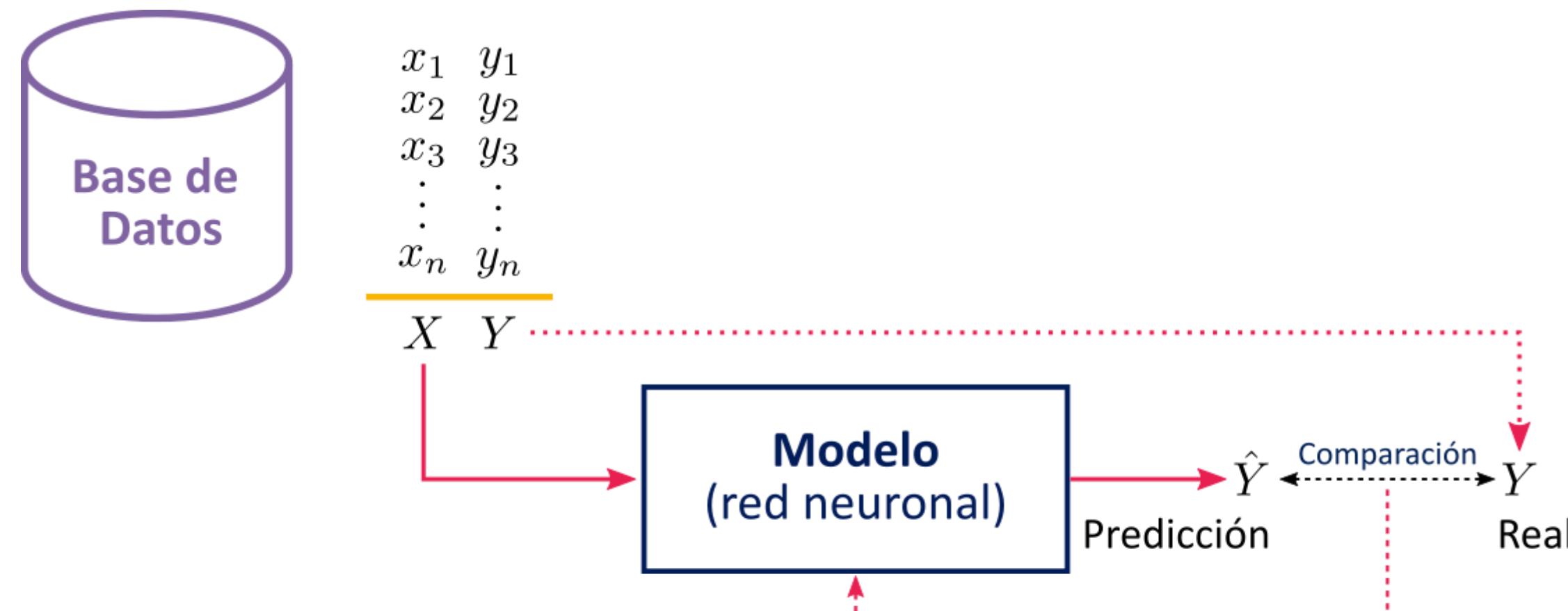
2.



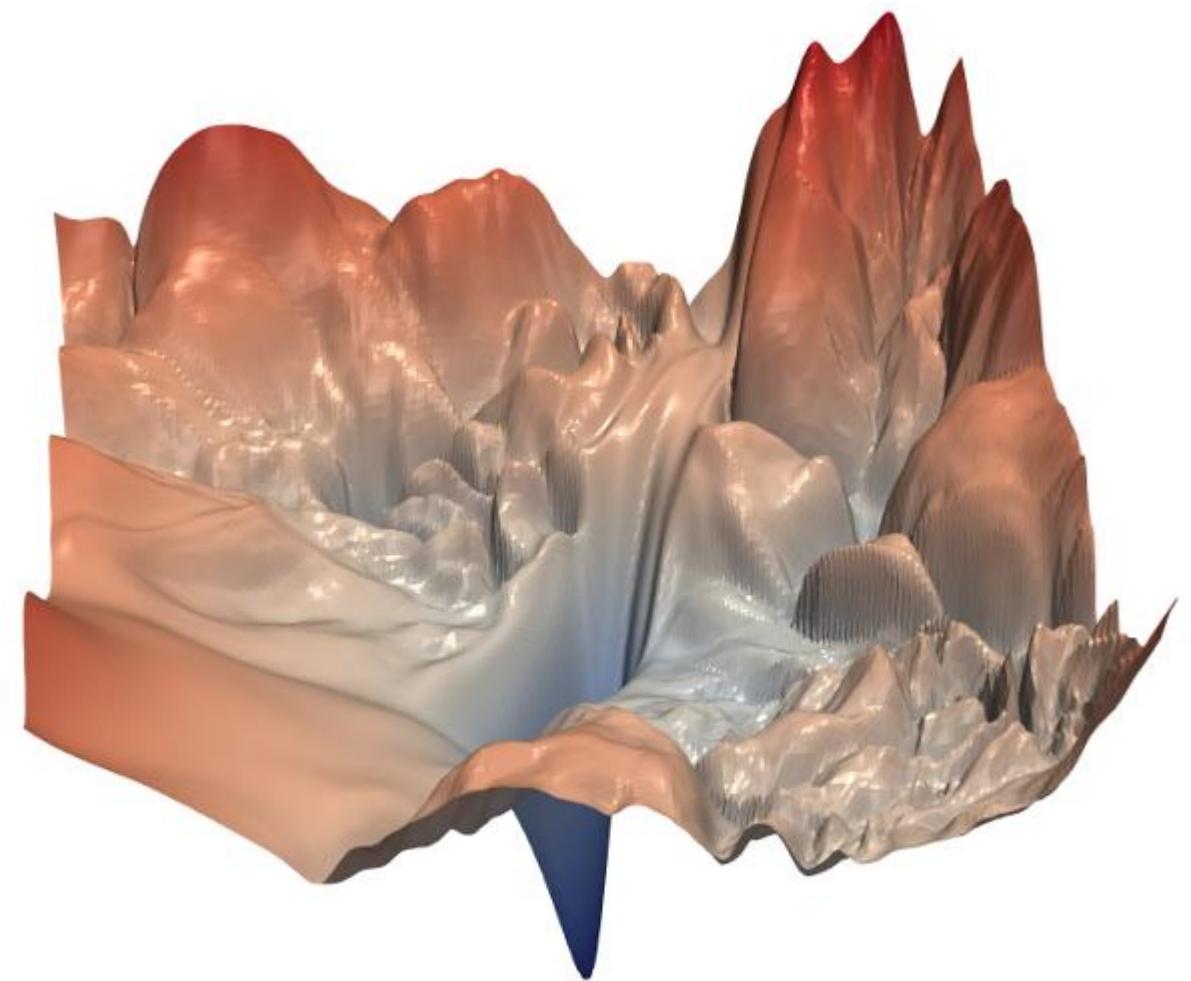
Training *tricks*



Train*ing*



Loss *landscape*



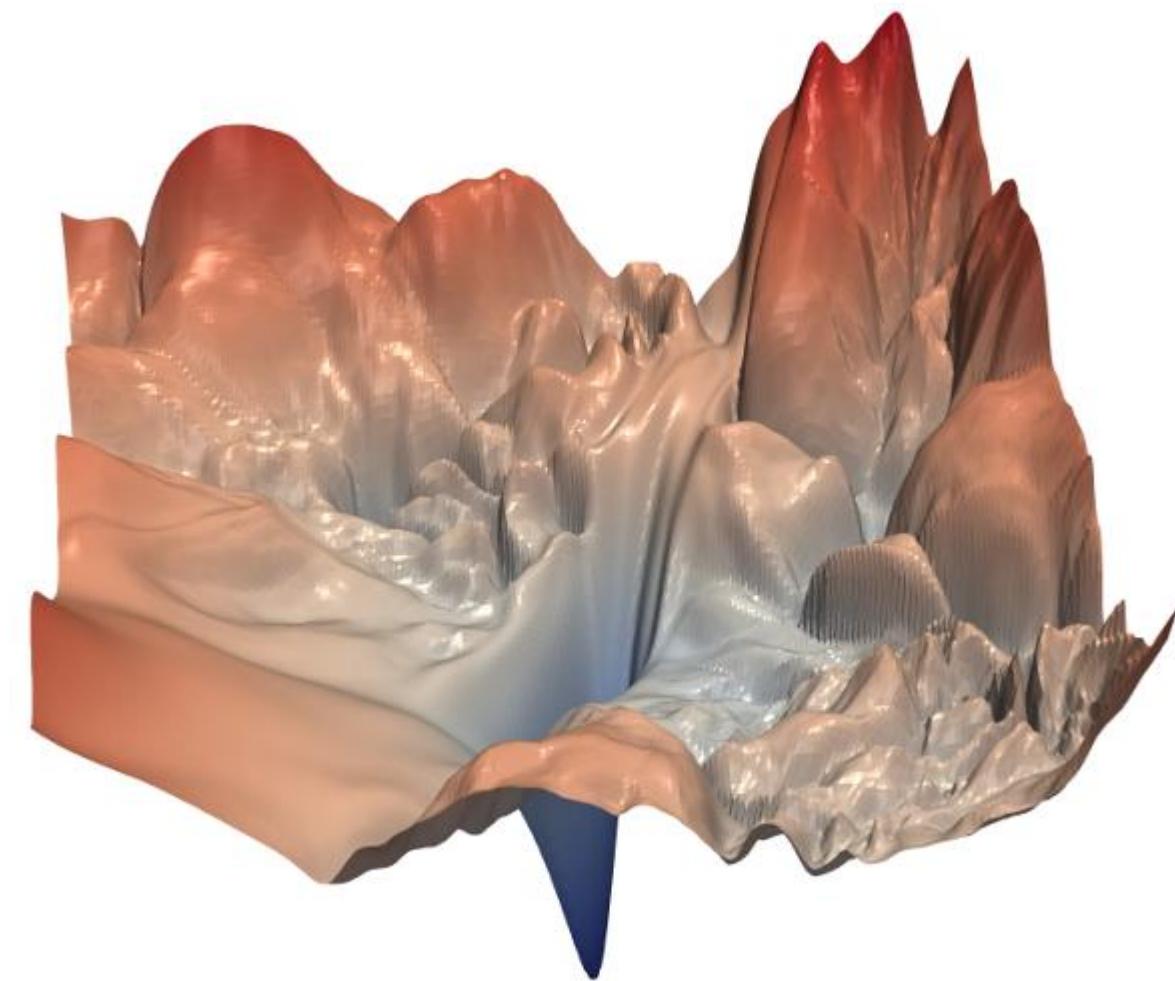
(a) without skip connections



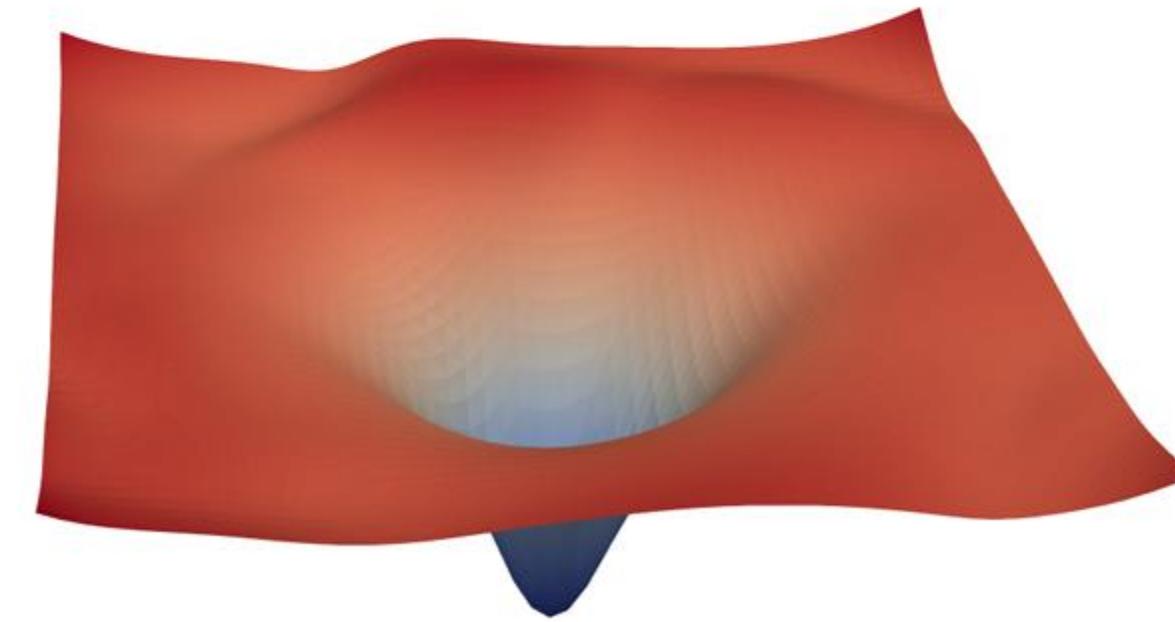
TRANSFORMATEC

Hao Li et al. (2017) "Visualizing the Loss Landscape of Neural Nets".
arXiv e-prints. arXiv preprint arXiv:1712.09913.

Loss *landscape*



(a) without skip connections



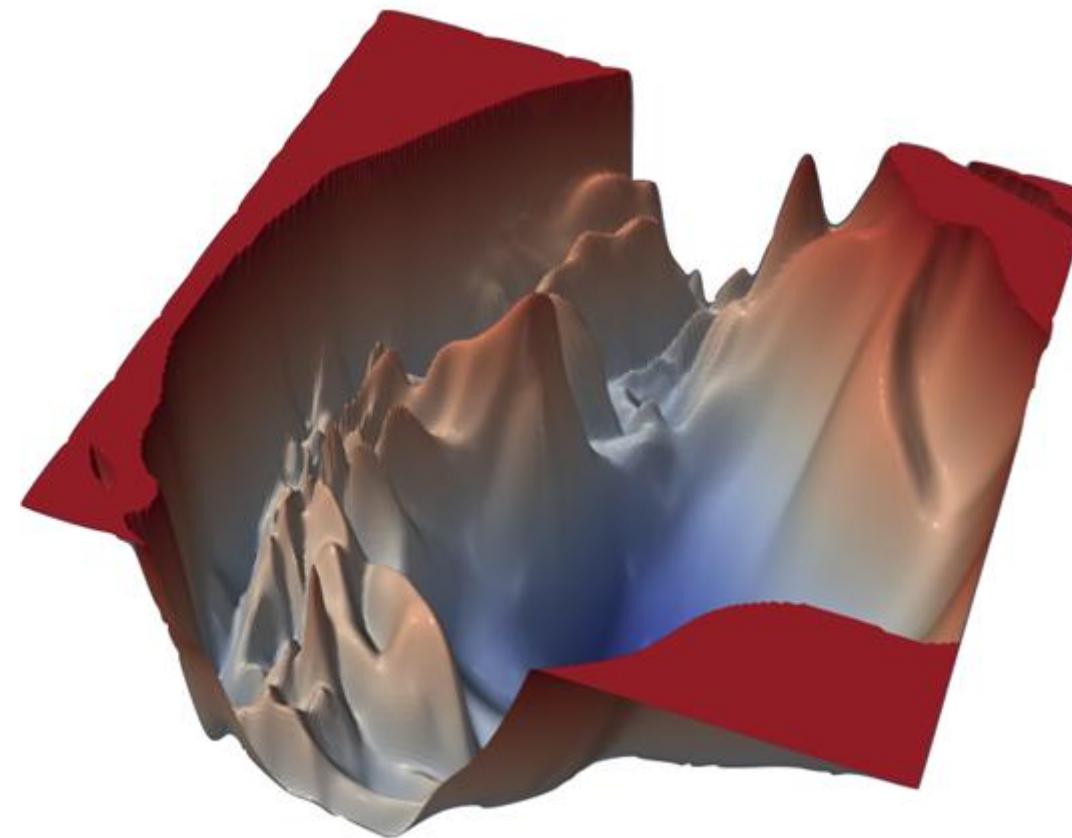
(b) with skip connections



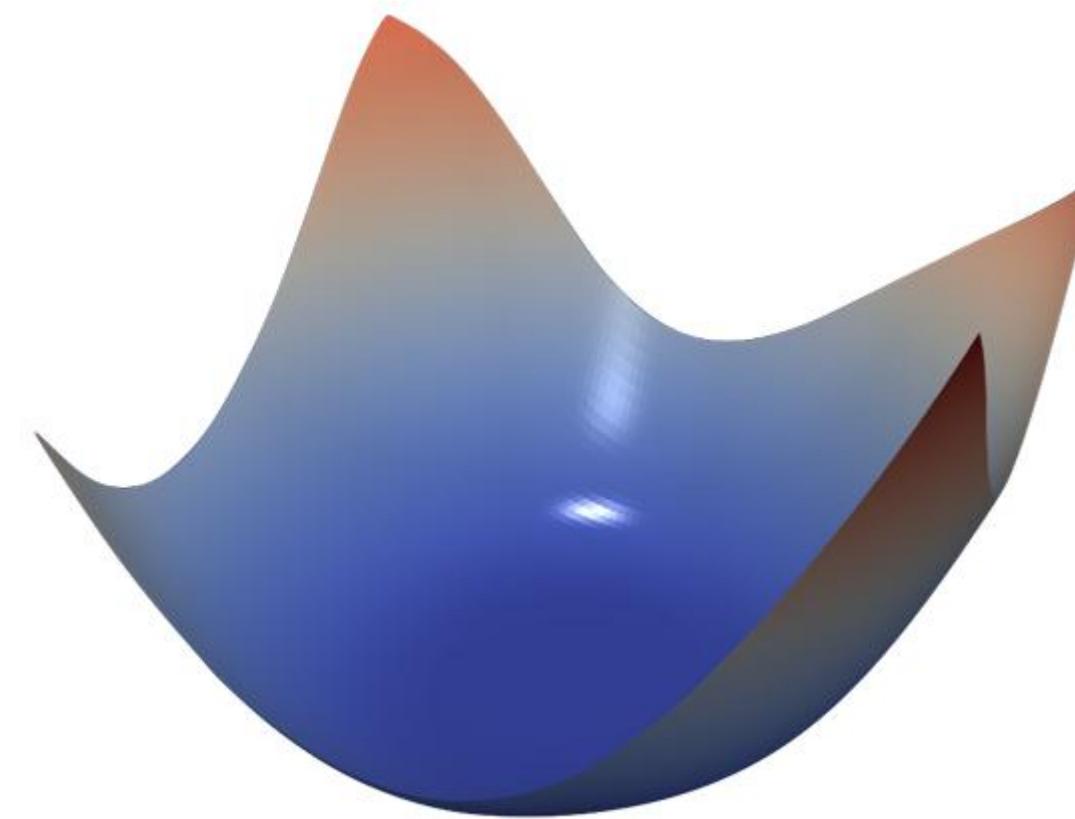
TRANSFORMATEC

Hao Li et al. (2017) "Visualizing the Loss Landscape of Neural Nets".
arXiv e-prints. arXiv preprint arXiv:1712.09913.

Loss *landscape*



(a) ResNet-110, no skip connections



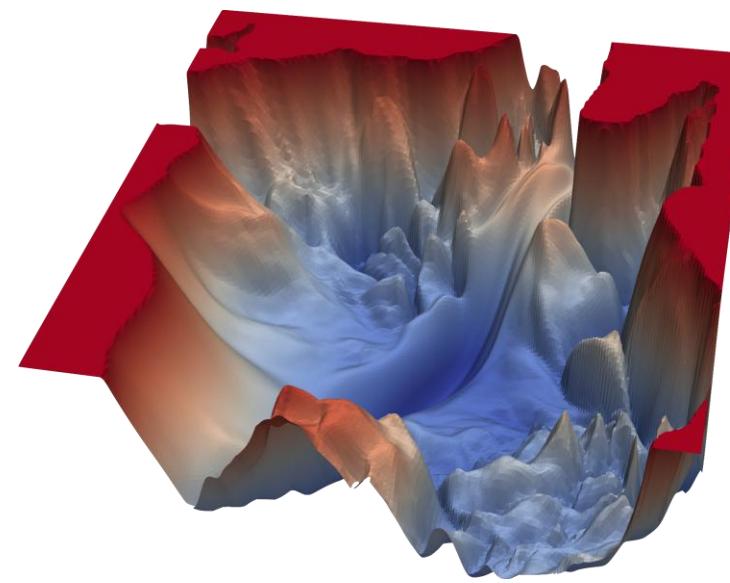
(b) DenseNet, 121 layers



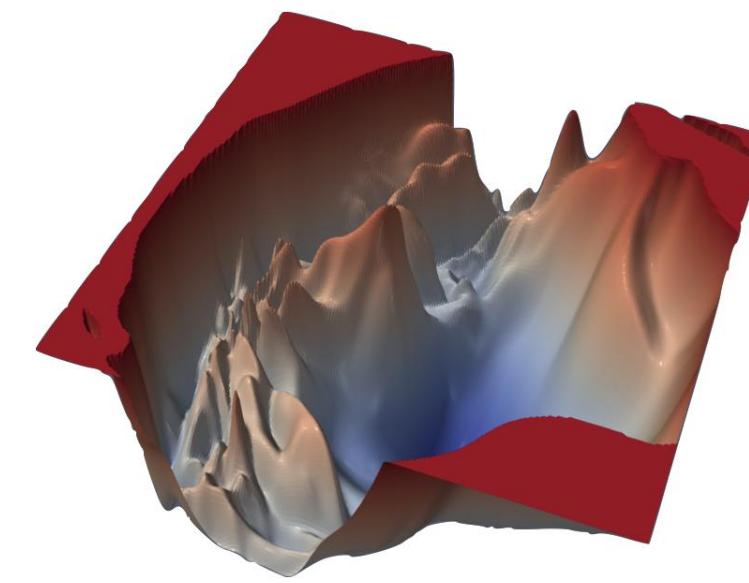
TRANSFORMATEC

Hao Li et al. (2017) "Visualizing the Loss Landscape of Neural Nets".
arXiv e-prints. arXiv preprint arXiv:1712.09913.

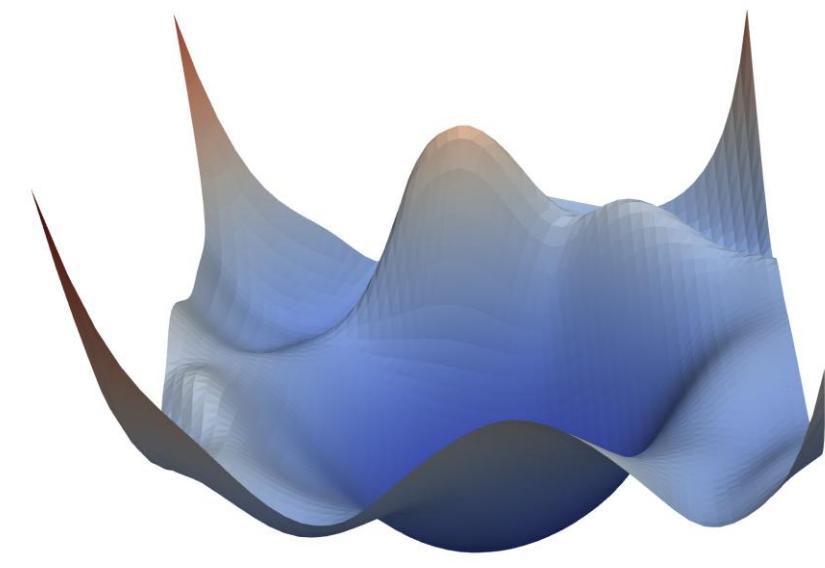
Loss *landscape*



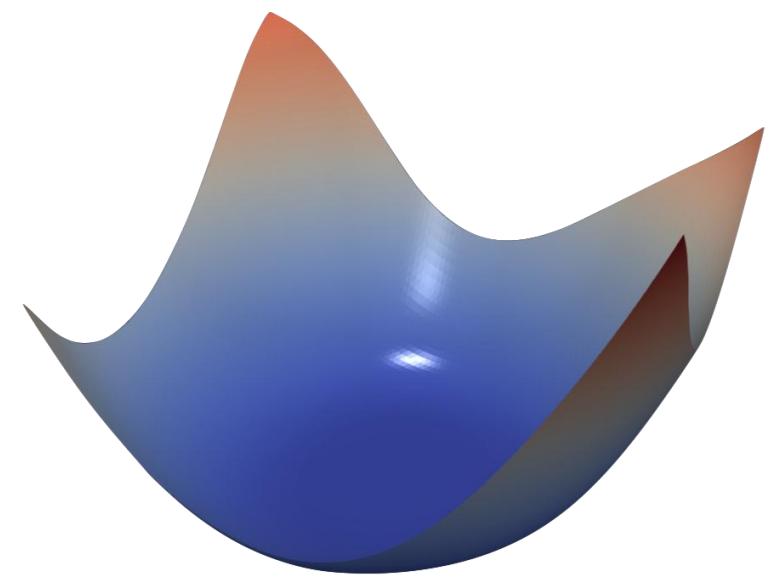
VGG-56



ResNet-110 without skip connections



Renset-56



Densenet-121



TRANSFORMATEC

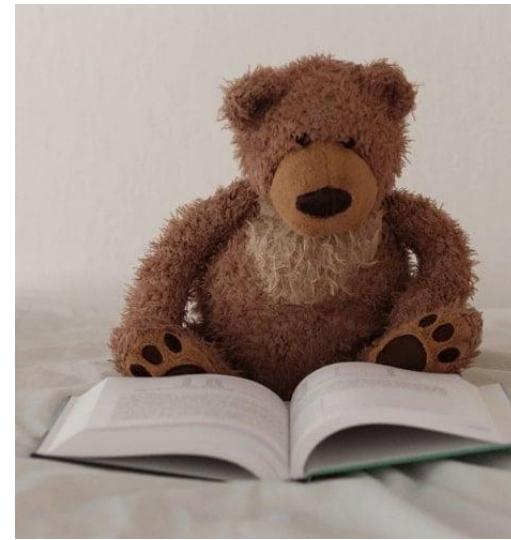
Hao Li et al. (2017) "Visualizing the Loss Landscape of Neural Nets".
arXiv e-prints. arXiv preprint arXiv:1712.09913.

Data augmentation

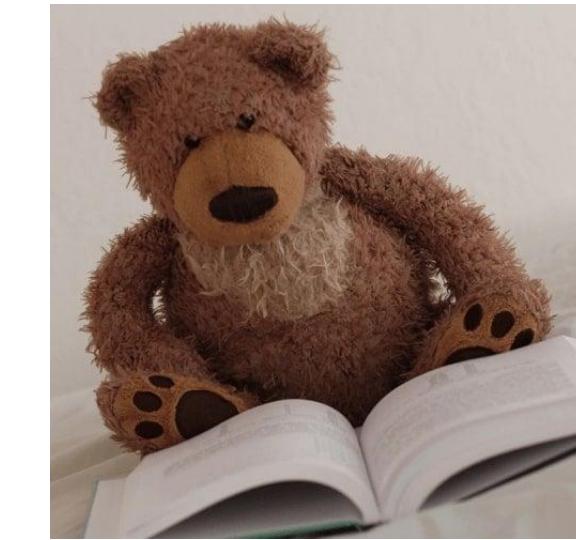
Original



Flip



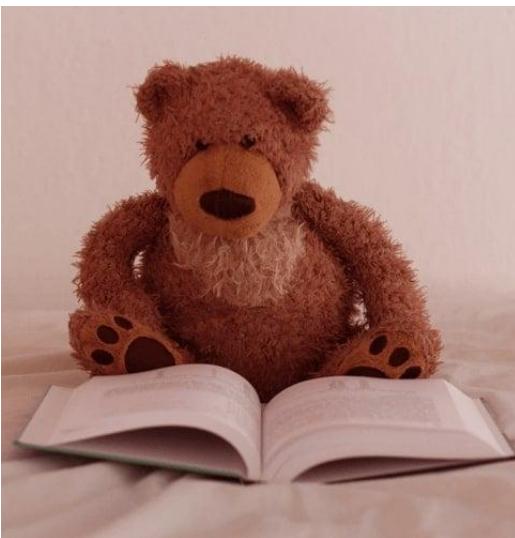
Rotation



Random crop



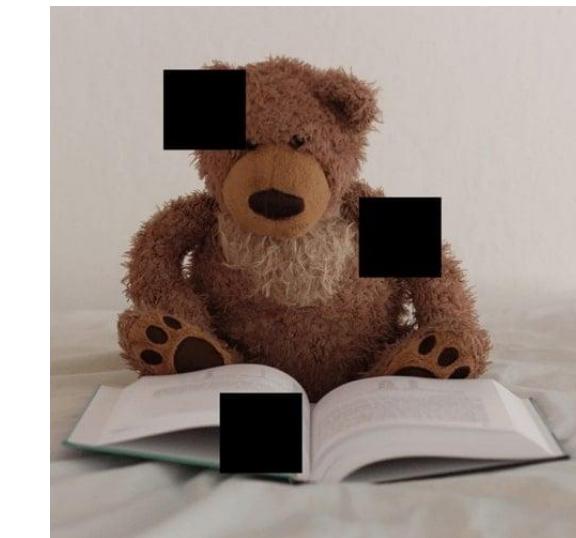
Color shift



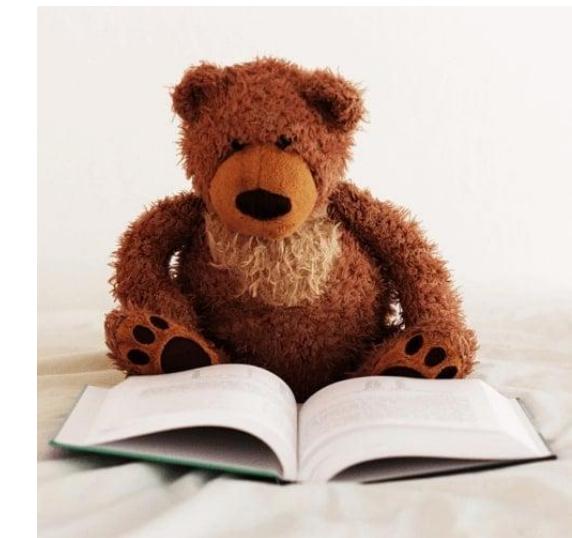
Noise addition



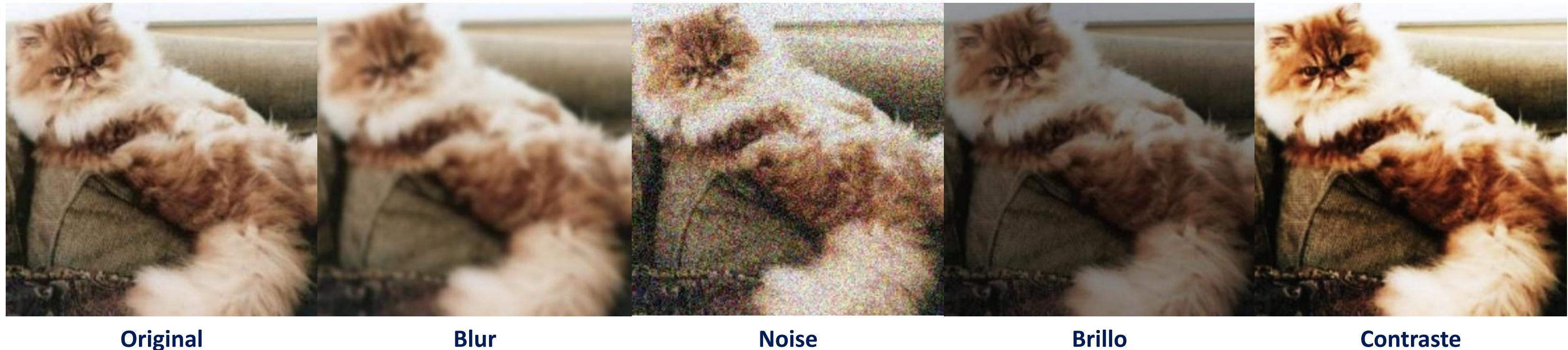
Information loss



Contrast change



Data augmentation



Original

Blur

Noise

Brillo

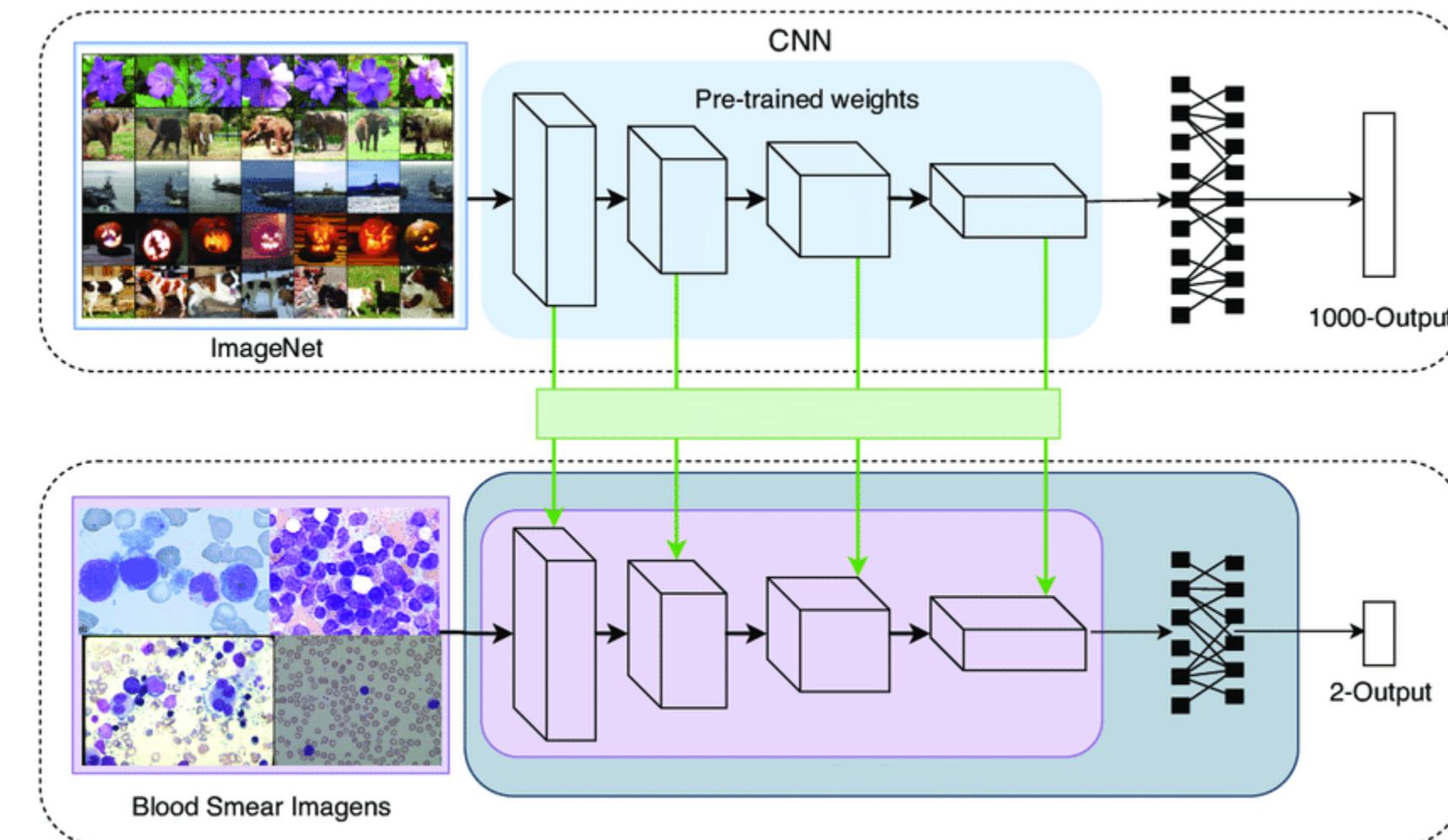
Contraste



Data *augmentation*



Pre-training



Pre-training

**Small
Dataset**

**Big
Dataset**

Transfer learning

Fine-tuning

ImageNet domain

Recopilar más datos

Entrenar desde cero

Not similar to ImageNet

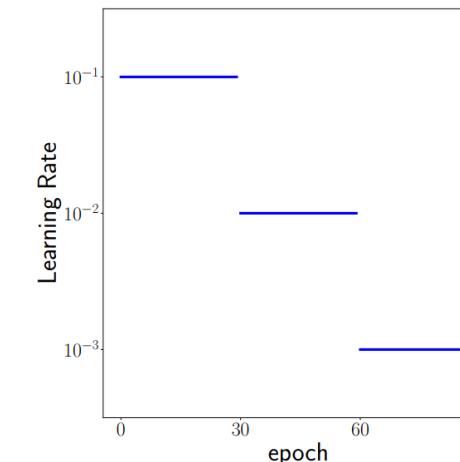


Learning *rate decay*

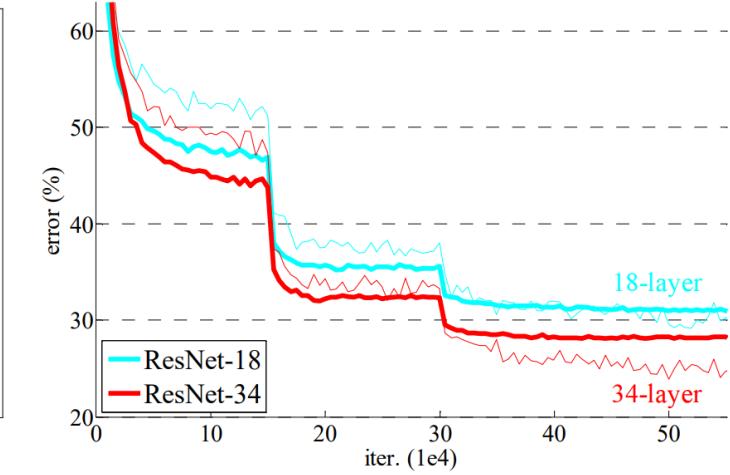
Step LR

$$\alpha = \alpha_0 \gamma^{\left[\frac{n}{s} \right]}$$

- n : época.
- s : Período de caída de la tasa de aprendizaje.
- γ : Factor multiplicativo del decaimiento de la tasa de aprendizaje.
- α_0 : taza de aprendizaje inicial.



(a) Learning rate decay strategy



(b) Figure taken from He et al. (2016)

Figure 1: Training error in (b) is shown by thin curves, while test error in (b) by bold curves.



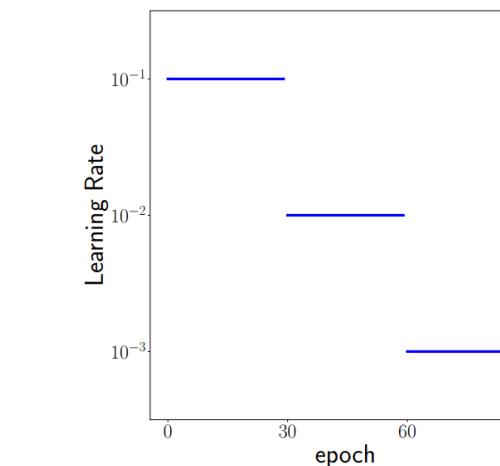
Learning *rate decay*

Step LR $\alpha = \alpha_0 \gamma^{\left\lceil \frac{n}{s} \right\rceil}$

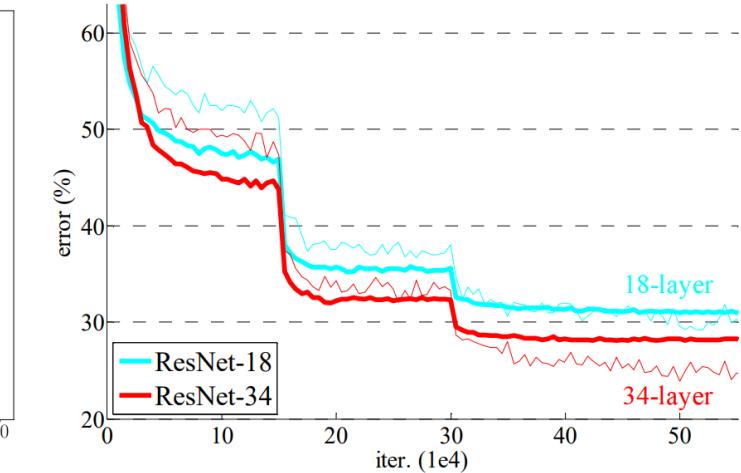
- n : época.
- s : Período de caída de la tasa de aprendizaje.
- γ : Factor multiplicativo del decaimiento de la tasa de aprendizaje.
- α_0 : tasa de aprendizaje inicial.

Decaimiento exponencial $\alpha = \alpha_0 e^{-kn}$

Decaimiento $1/t$ $\alpha = \alpha_0 \frac{1}{1 + kn}$



(a) Learning rate decay strategy



(b) Figure taken from He et al. (2016)

Figure 1: Training error in (b) is shown by thin curves, while test error in (b) by bold curves.



Learning *rate decay*

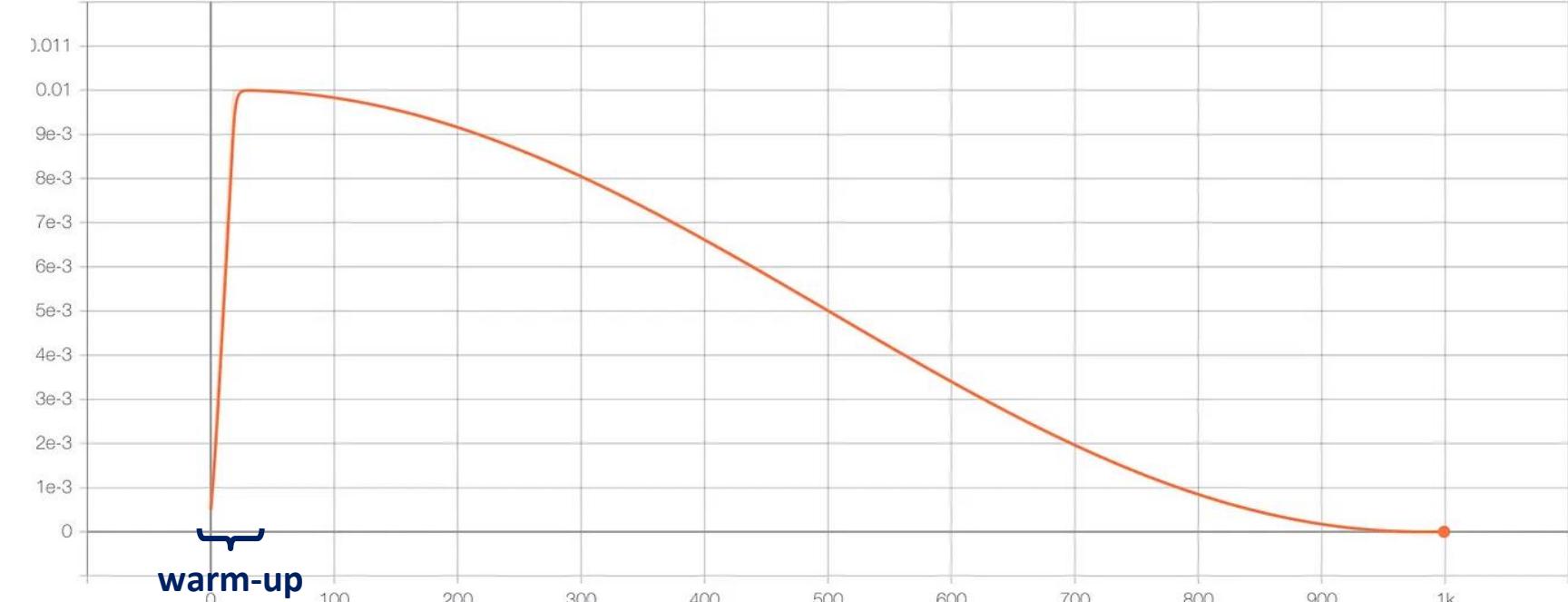
Step LR $\alpha = \alpha_0 \gamma^{\left[\frac{n}{s}\right]}$

- n : época.
- s : Período de caída de la tasa de aprendizaje.
- γ : Factor multiplicativo del decaimiento de la tasa de aprendizaje.
- α_0 : tasa de aprendizaje inicial.

Decaimiento exponencial $\alpha = \alpha_0 e^{-kn}$

Decaimiento $1/t$ $\alpha = \alpha_0 \frac{1}{1 + kn}$

Decaimiento coseno $\alpha = \alpha_0 \cdot \frac{1}{2} \left(1 + \cos \left(\frac{n}{N} \pi \right) \right)$



Bag of Tricks

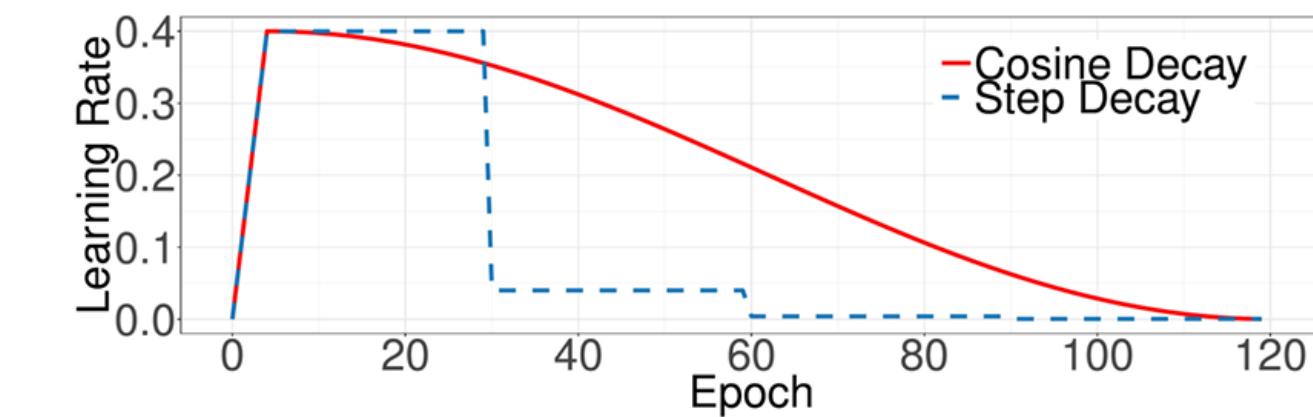
Cosine Learning Rate Decay

$$\eta_t = \eta_{\min} + \frac{1}{2} (\eta_{\max} - \eta_{\min}) \left(1 + \cos \left(k \frac{\pi}{\tau} \right) \right) \eta$$

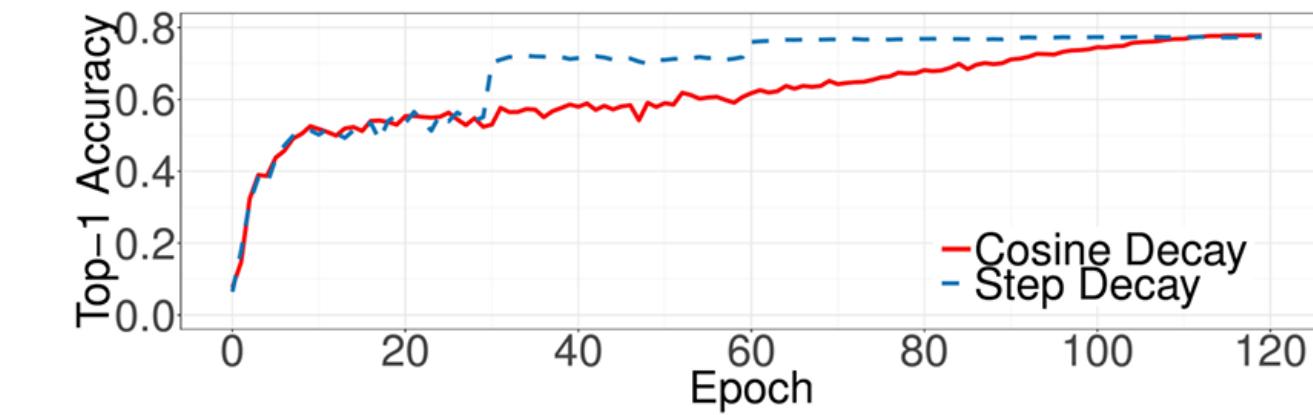
- η_t es la tasa de aprendizaje en el paso t .
- η_{\min} es el valor mínimo de la tasa de aprendizaje.
- η_{\max} es el valor máximo de la tasa de aprendizaje.
- τ es el número total de epochs.
- k es el número de epochs transcurridos hasta el momento actual.



Learning Rate Schedule



Validation Accuracy



Bag of Tricks

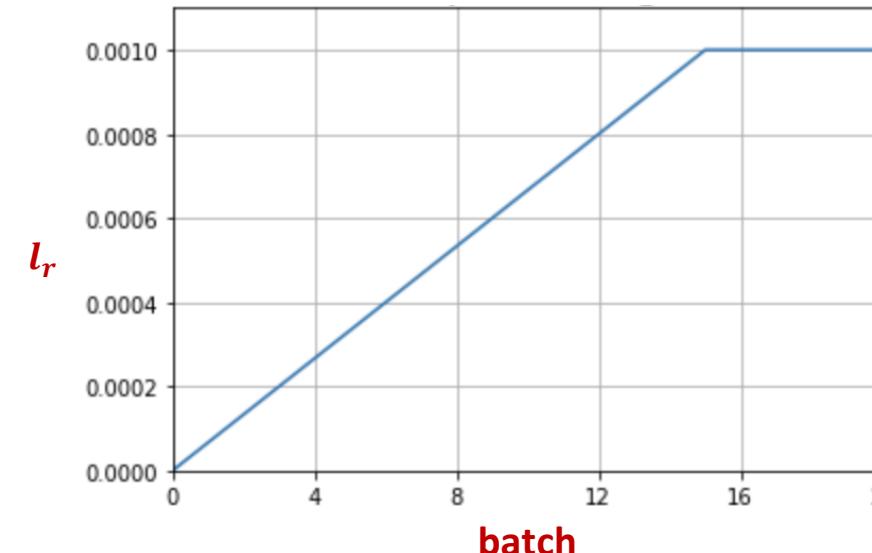
Large-batch training

Linear scaling learning rate

Se propone aumentar la tasa de aprendizaje de manera proporcional al tamaño del *batch*.

$$l_r = 0.1 \frac{b}{256}$$

Learning rate warmup



Zero γ

El parámetro de escala γ se inicializa en 0:

$$\text{BN}_{\gamma,\beta}(z_i) = \gamma \hat{z}_i + \beta$$

Esto hace que, al inicio del entrenamiento, la salida de esos bloques sea esencialmente la misma que la entrada (es decir, los bloques no aportan cambios significativos a los datos que procesan).

No bias decay

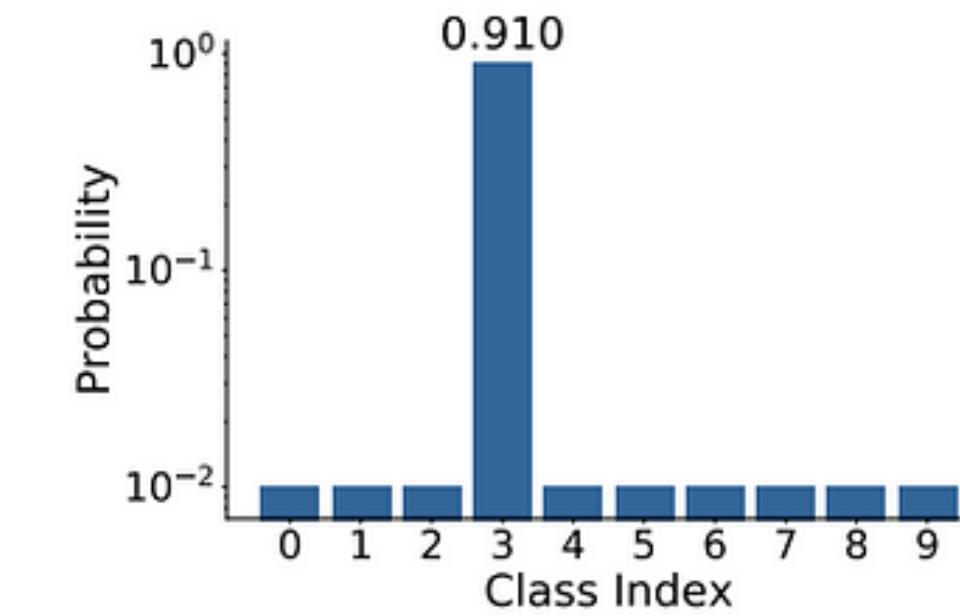
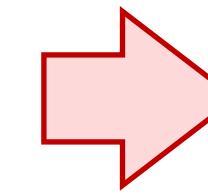
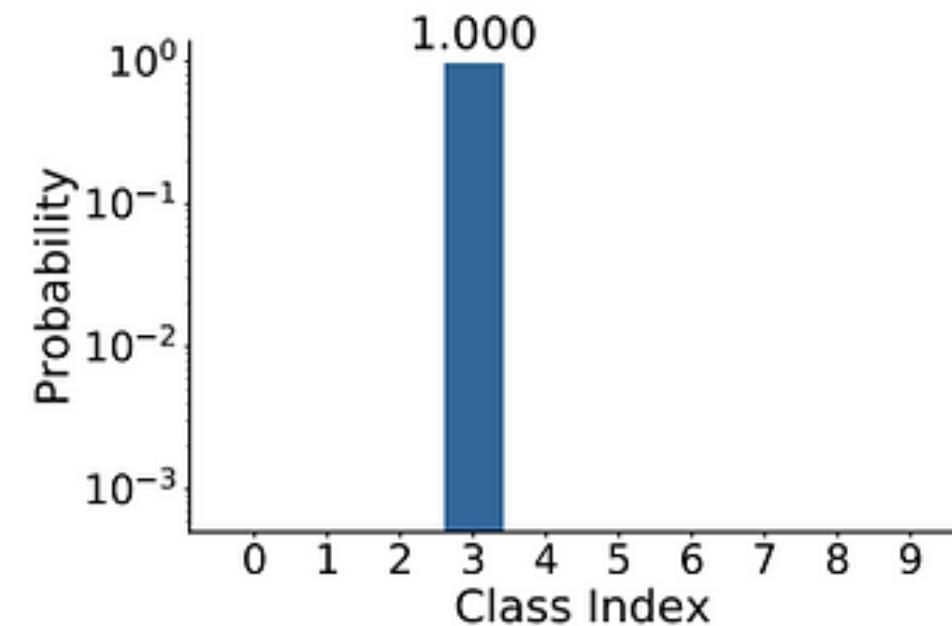
$$\text{Weight decay: } g_t = \nabla_{\theta} \mathcal{L}(\theta_t) + \lambda \theta_t$$

Se recomienda aplicar el *weight decay* solo a los pesos de las capas convolucionales y *fully-connected*, excluyendo los *biases* y los parámetros γ y β de las capas BN. Esto reduce el riesgo de sobreajuste y mejora la generalización.

Bag of *Tricks*

Label Smoothing

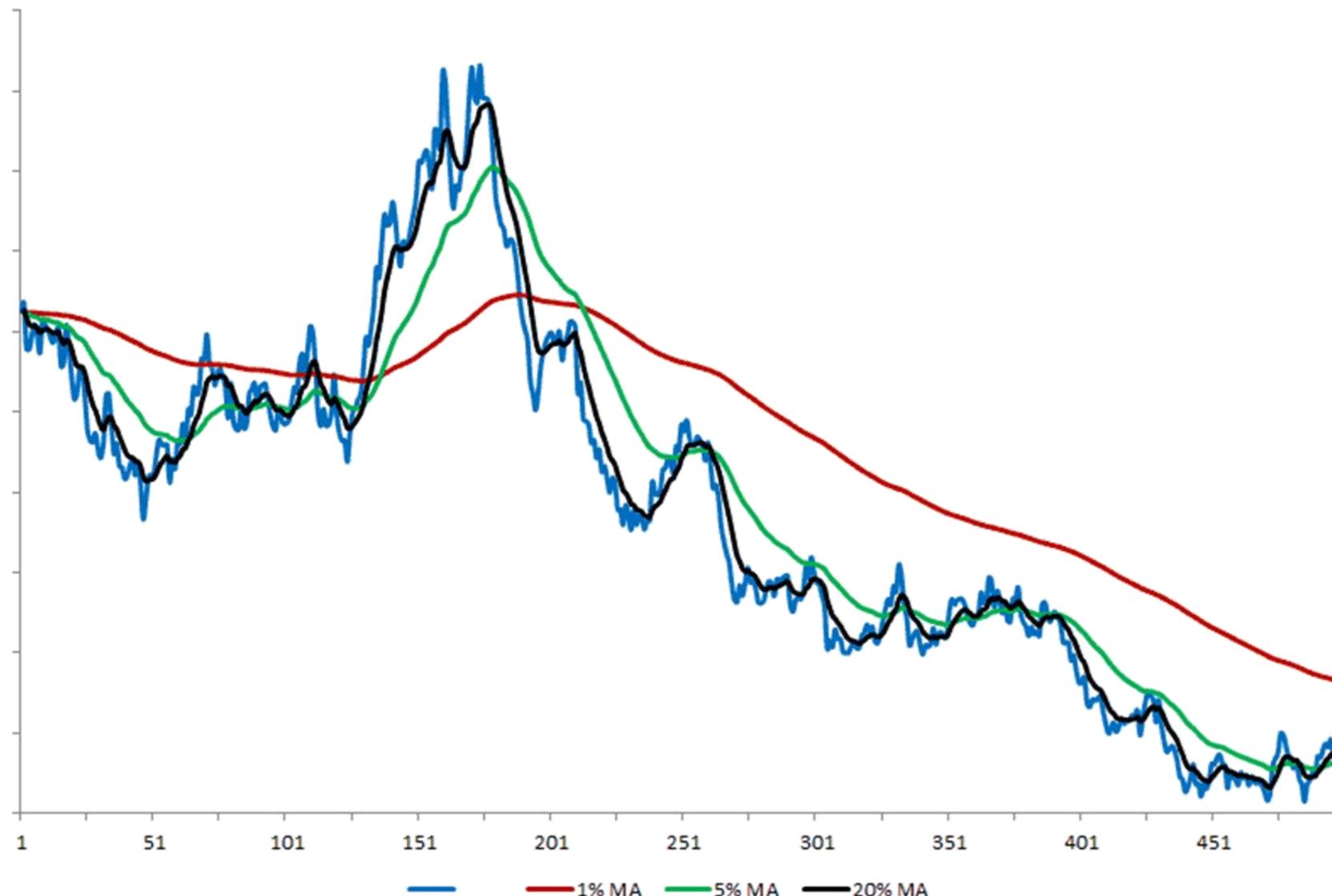
$$p_i = \begin{cases} 1 - \varepsilon, & i = y \\ \frac{\varepsilon}{K - 1}, & i \neq y \end{cases}$$



ResNet-RS

(Re-Scaling ResNet)

Exponential moving average (EMA) of the weights



$$\theta_t^{\text{EMA}} = \alpha \theta_{t-1}^{\text{EMA}} + (1 - \alpha) \theta_t$$

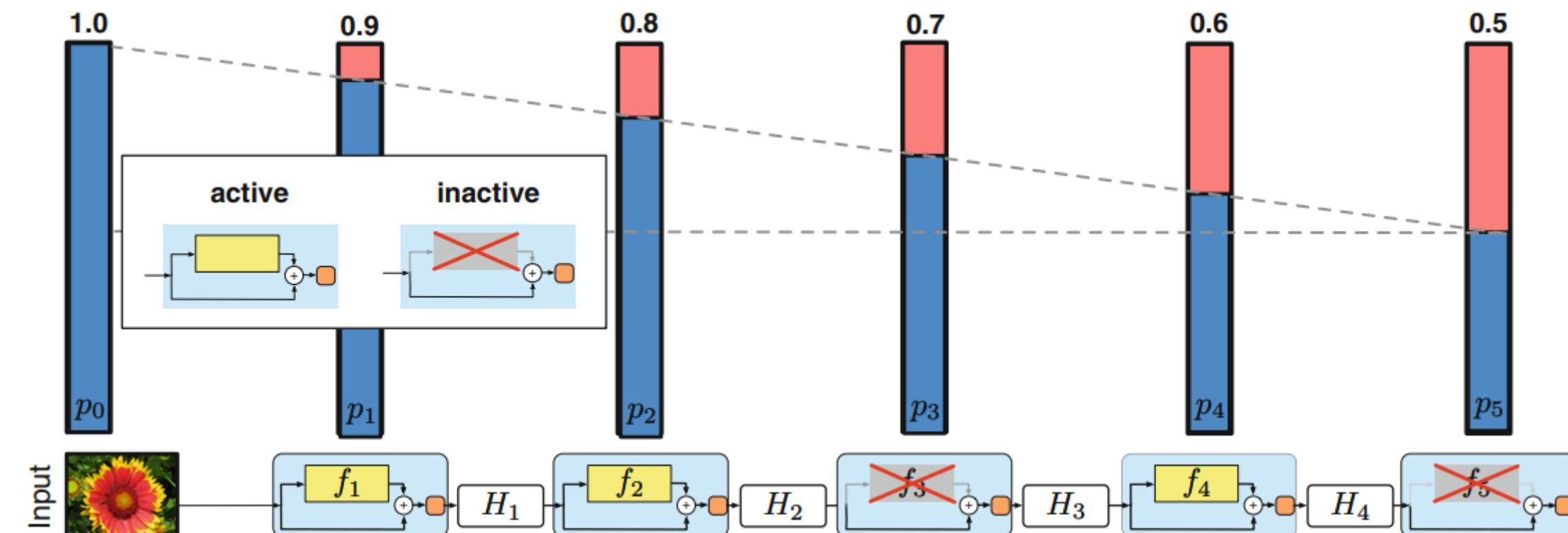
- θ_t^{EMA} es el valor del parámetro suavizado en el paso t .
- α es el factor de suavizado, que controla la velocidad con la que se "olvidan" los valores antiguos. Típicamente, α se establece en un valor cercano a 1, como 0.999.
- θ_t es el valor actual del parámetro en el paso t .



ResNet-RS

(Re-Scaling ResNet)

Stochastic Depth



$$p_\ell = 1 - \frac{\ell}{L}(1 - p_L)$$

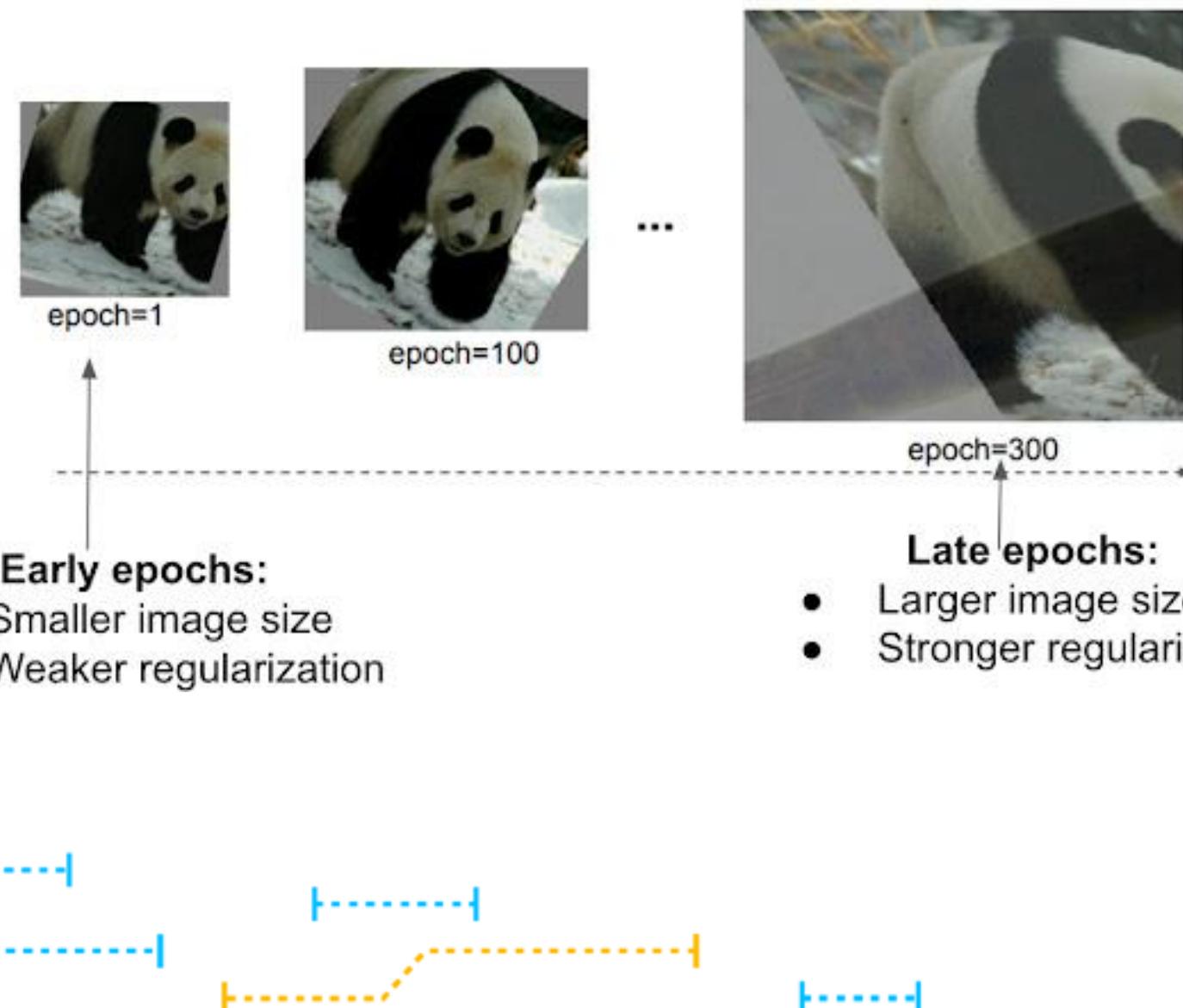


TRANSFORMATEC

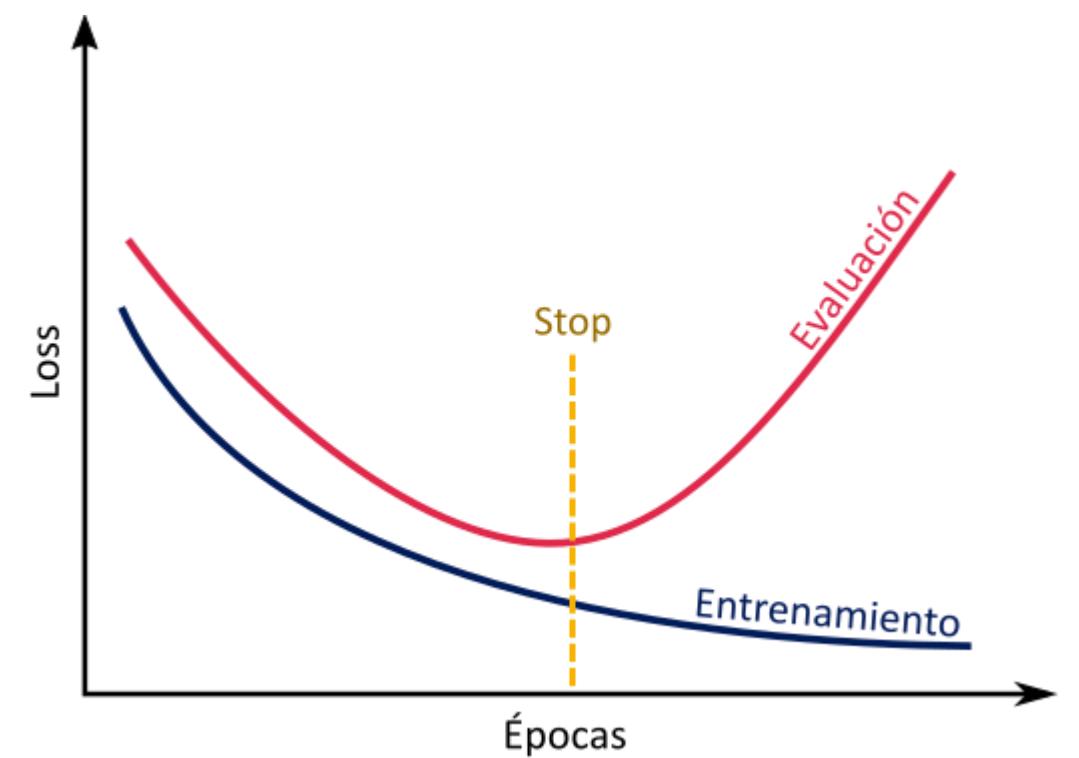
Gao Huang et al. (2016) "Deep networks with stochastic depth".
Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 646–661). Springer International Publishing.

EfficientNetV2

Progressive Learning



Early stopping



Early stopping

Valor de Loss de entrenamiento: \mathcal{L}_{tr}

Valor de Loss de evaluación: \mathcal{L}_{ev}

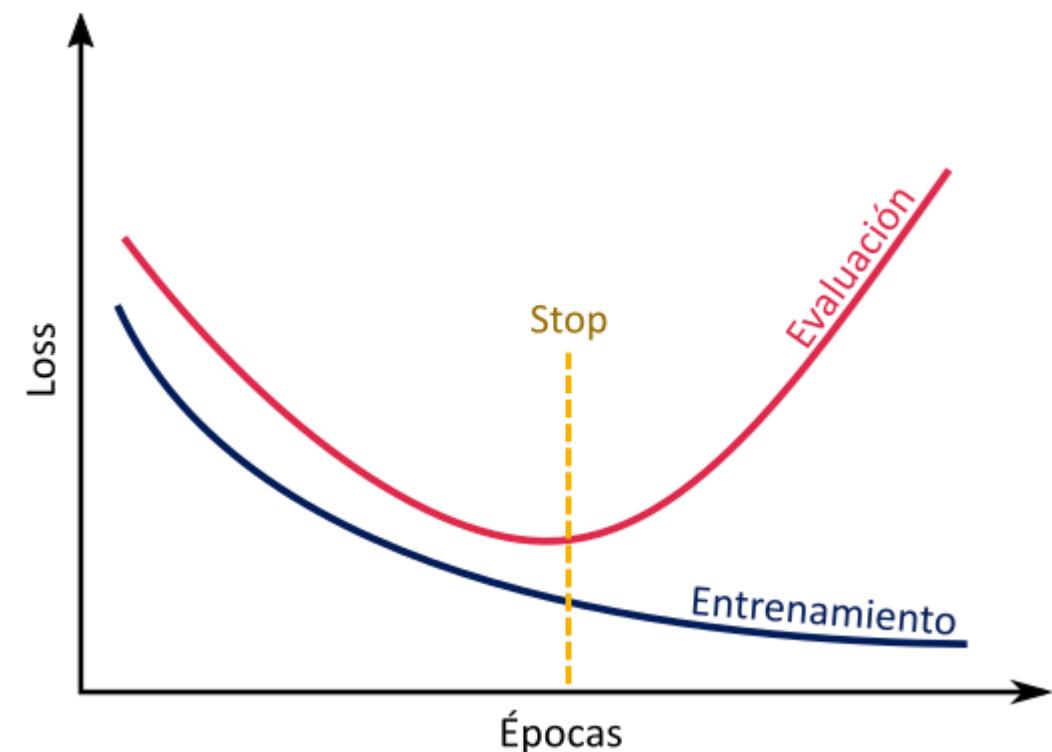
Mejor valor obtenido de \mathcal{L}_{ev} :

$$\mathcal{L}_{op} = \min_{t' \leq t} \mathcal{L}_{ev}(t')$$

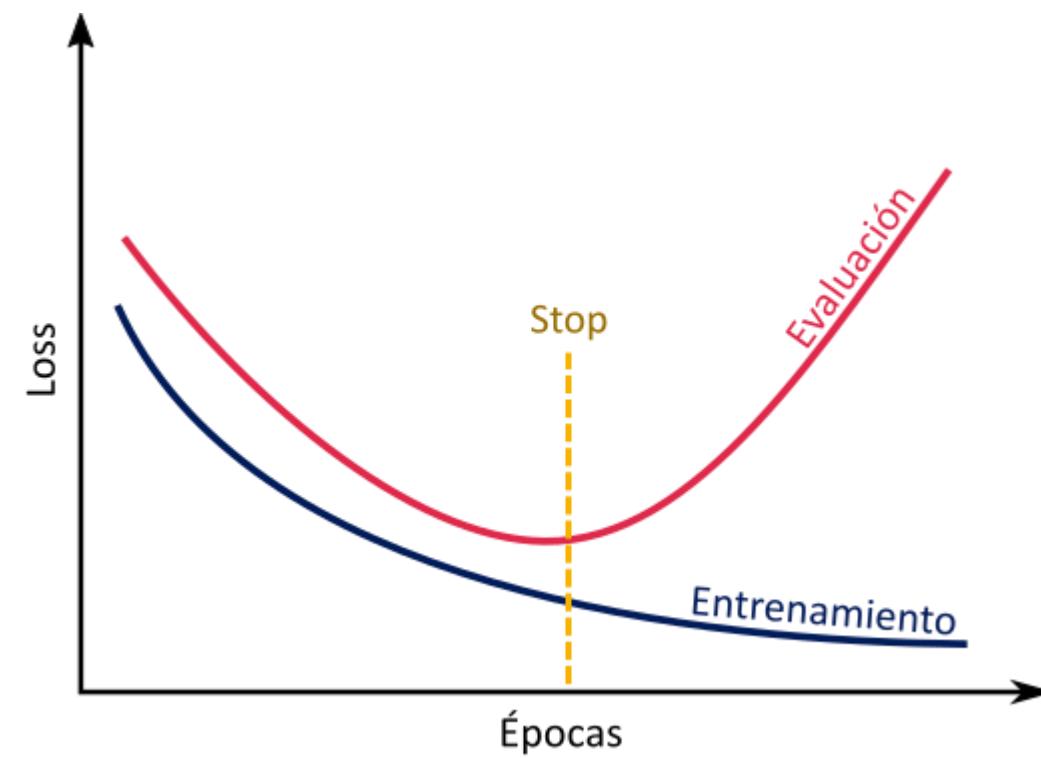
Criterio 1:

$$GL(t) = 100 \left(\frac{\mathcal{L}_{ev}(t)}{\mathcal{L}_{op}} - 1 \right)$$

GL_α : Parada después de la primera época t con $GL(t) > \alpha$



Early stopping



Valor de Loss de entrenamiento: \mathcal{L}_{tr}

Valor de Loss de evaluación: \mathcal{L}_{ev}

Mejor valor obtenido de \mathcal{L}_{ev} :

$$\mathcal{L}_{op} = \min_{t' \leq t} \mathcal{L}_{ev}(t')$$

Criterio 1:

$$GL(t) = 100 \left(\frac{\mathcal{L}_{ev}(t)}{\mathcal{L}_{op}(t)} - 1 \right)$$

GL_α : Parada después de la primera época t con $GL(t) > \alpha$

Criterio 2:

$$P_k(t) = 1000 \left(\frac{\mu_k(t)}{\delta_k(t)} - 1 \right)$$

donde: $\mu_k(t) = \frac{1}{k} \sum_{t'=t-k+1}^t \mathcal{L}_{tr}(t')$,

$$\delta_k(t) = \min_{t' \in [t-k+1, t]} \mathcal{L}_{tr}(t')$$

PQ_α : Parada después de la primera época de final de franja t con $\frac{GL(t)}{P_k(t)} > \alpha$



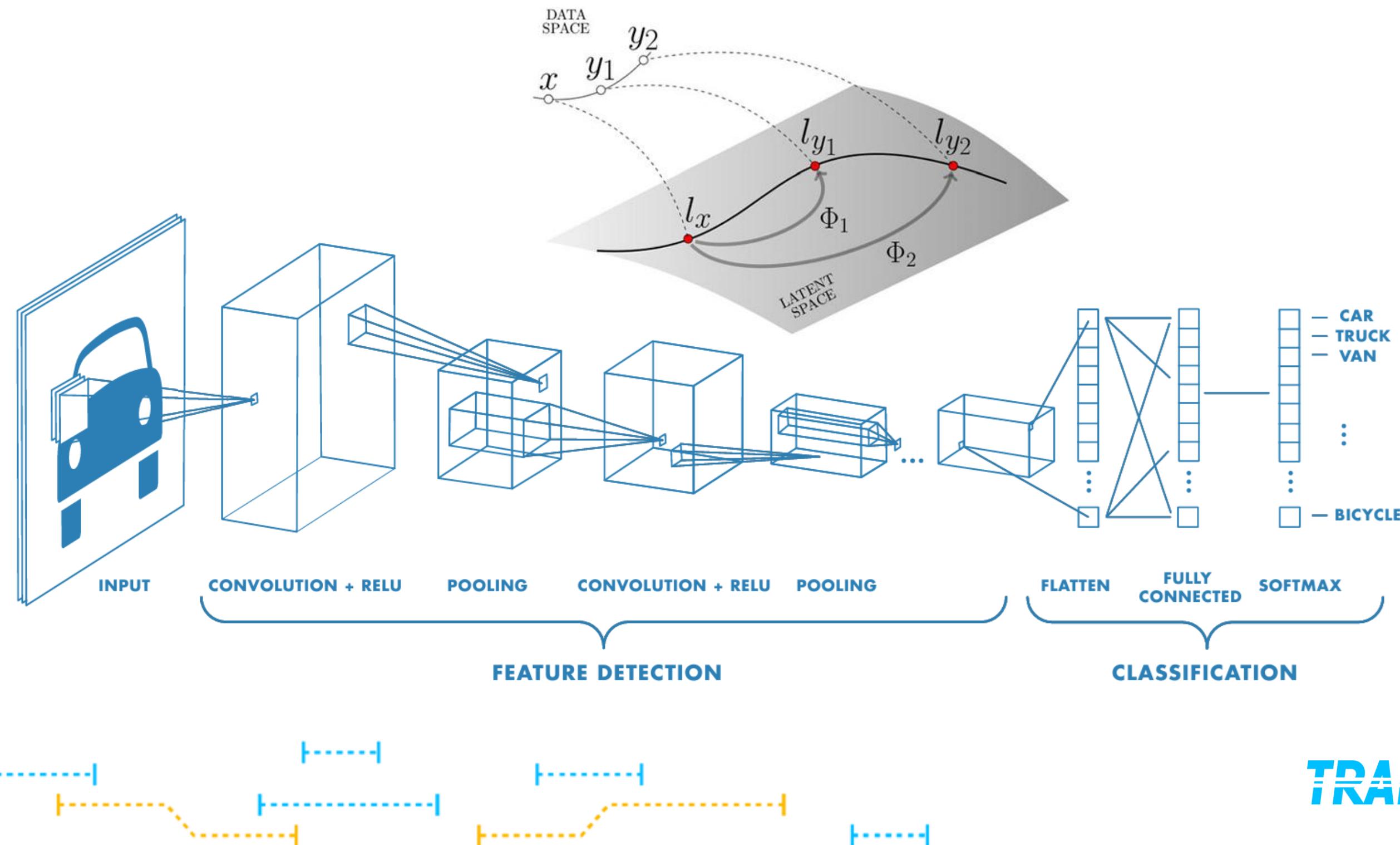
3.

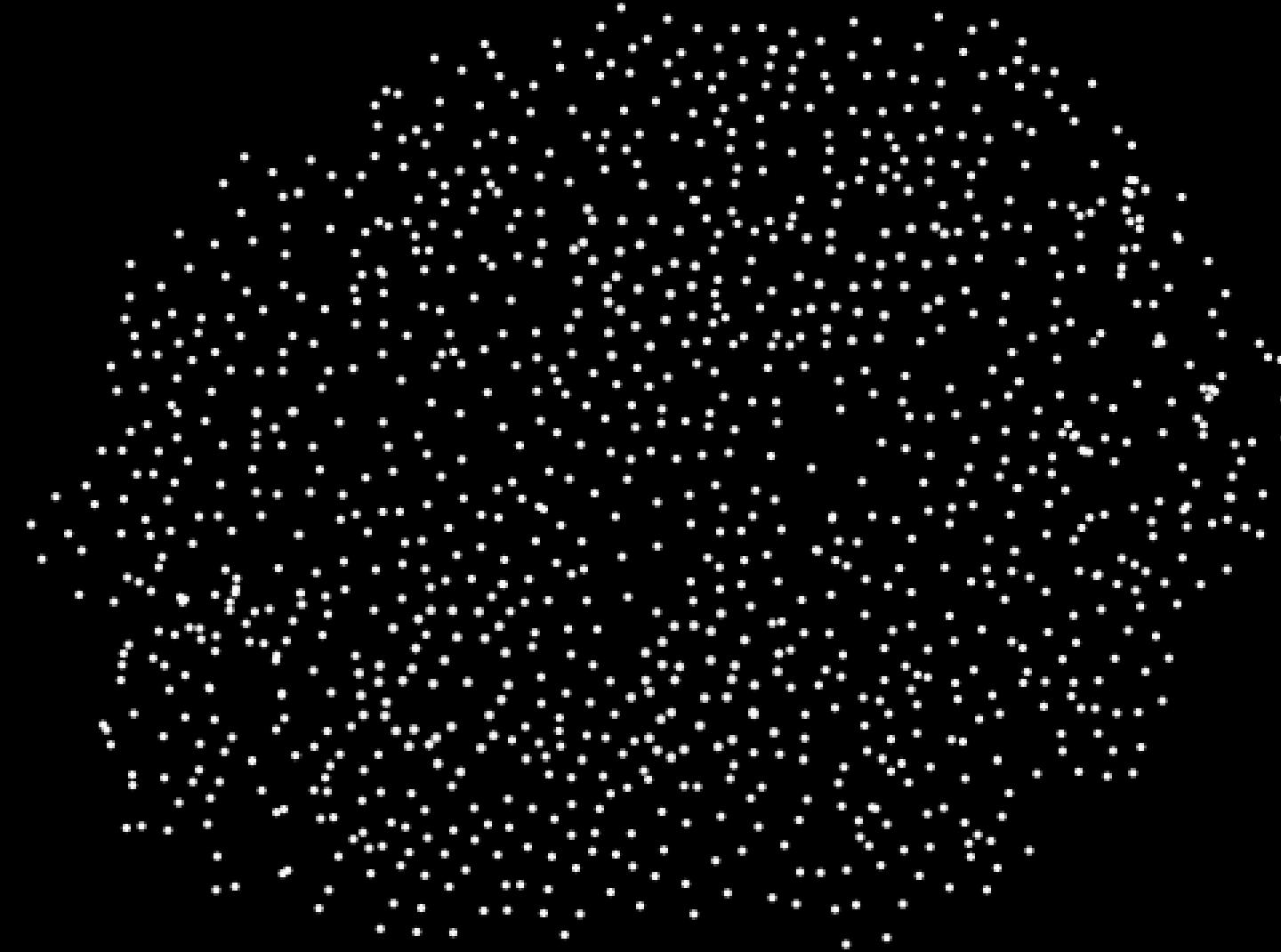


Learning *representation*

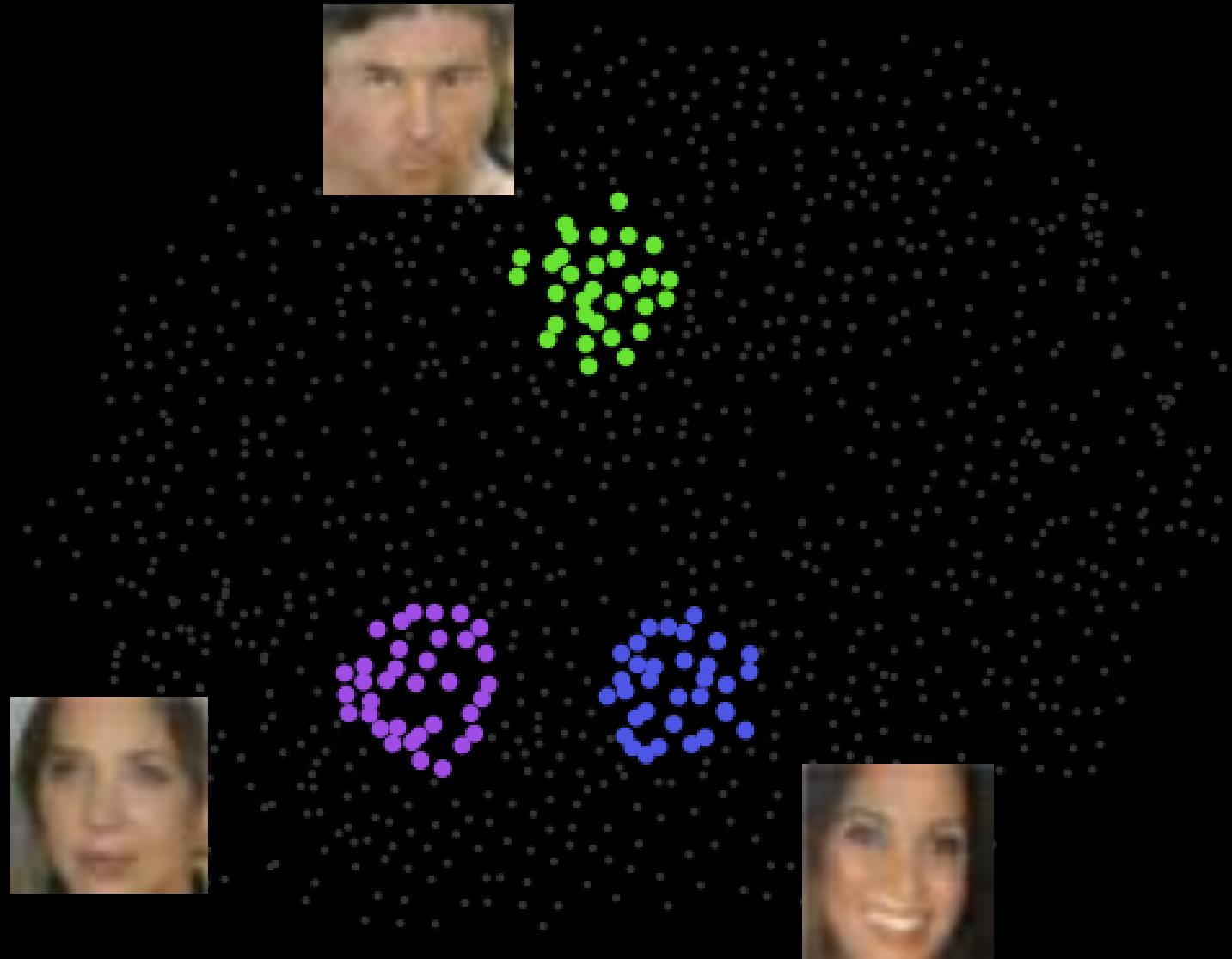


Deep learning pipeline



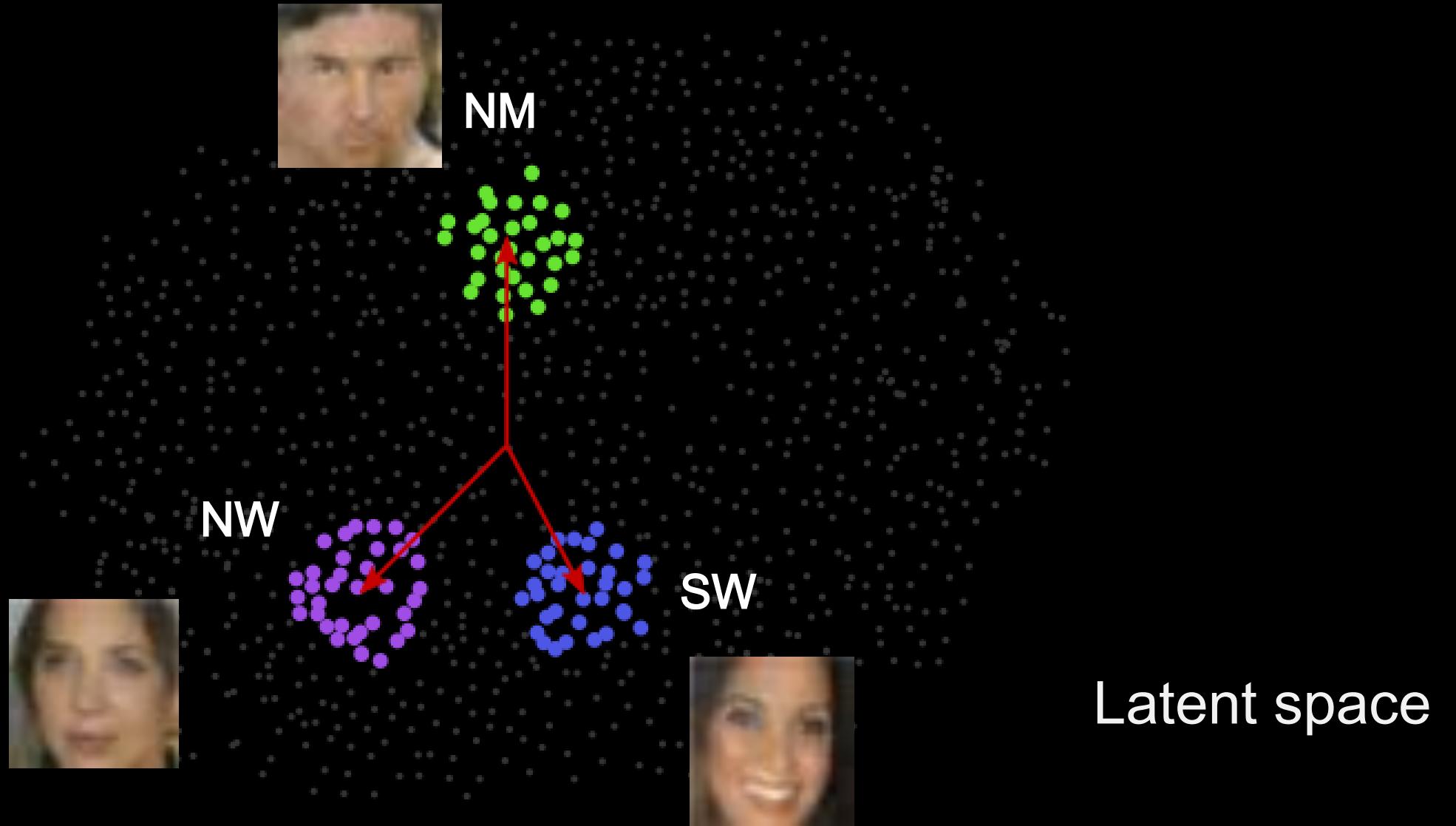


Latent space

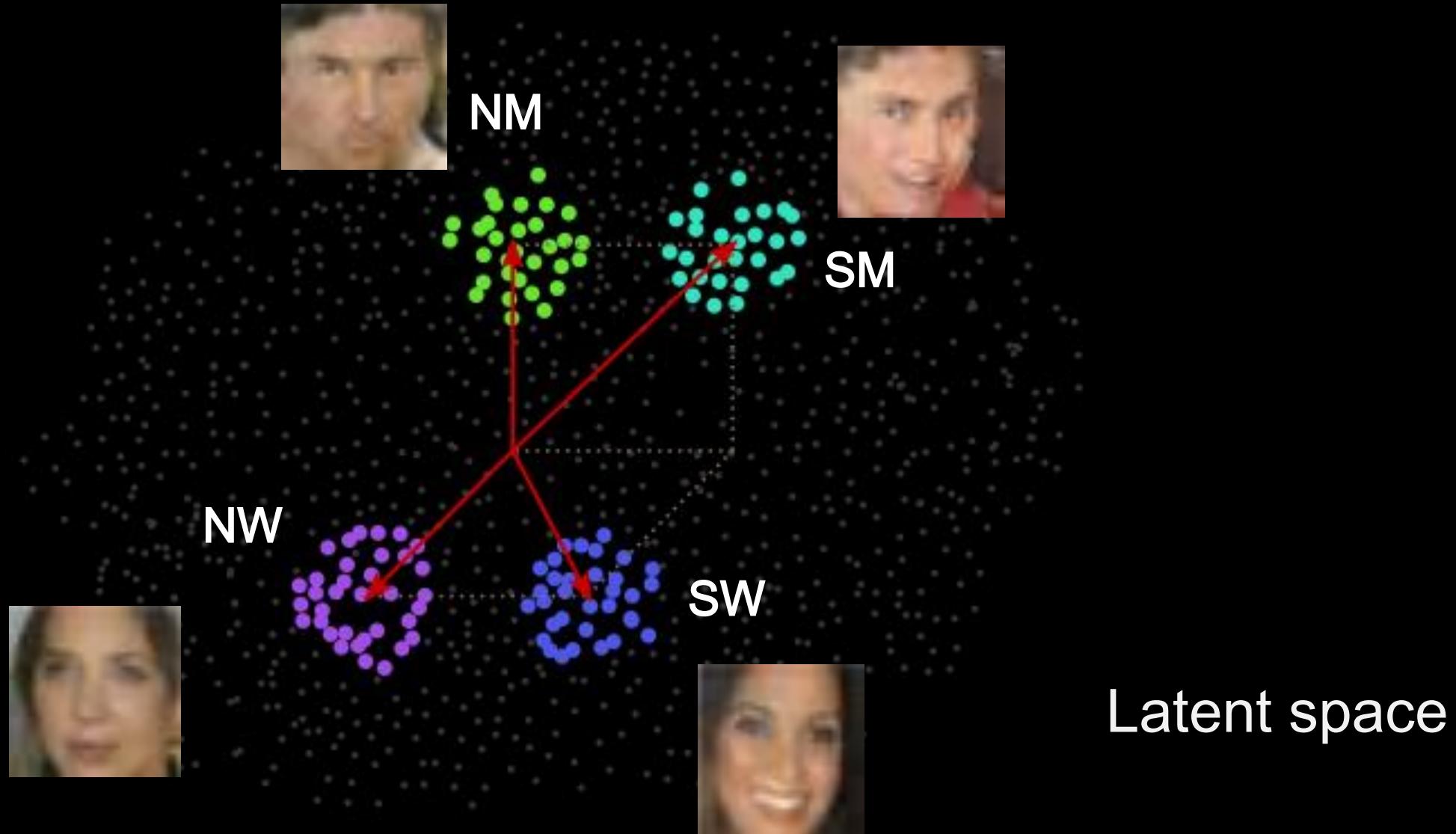


Latent space

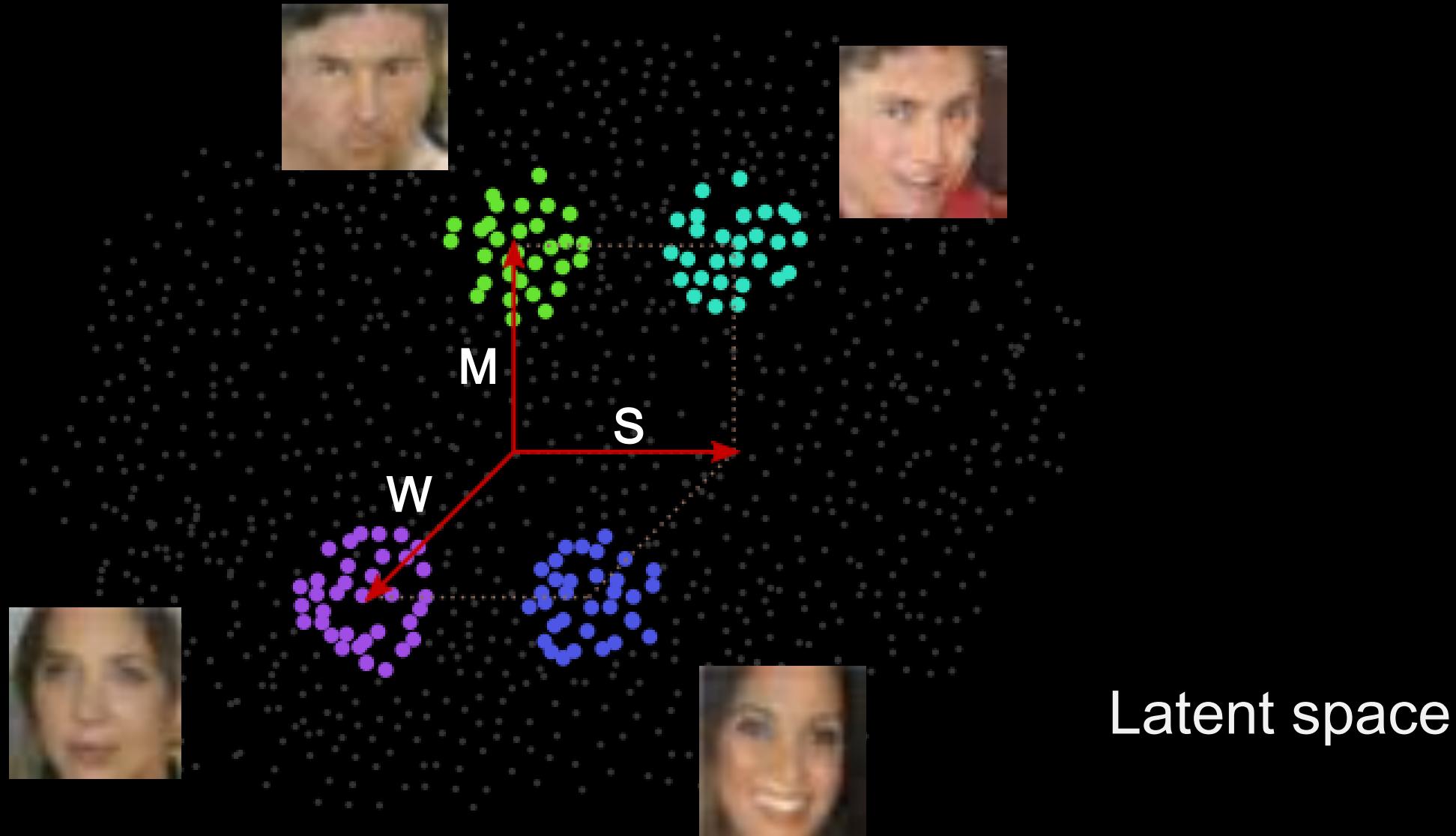
Alec Radford et al. (2016) "Unsupervised representation learning with deep convolutional generative adversarial networks".
4th International Conference on Learning Representations (ICRL) , Conference Track Proceedings.



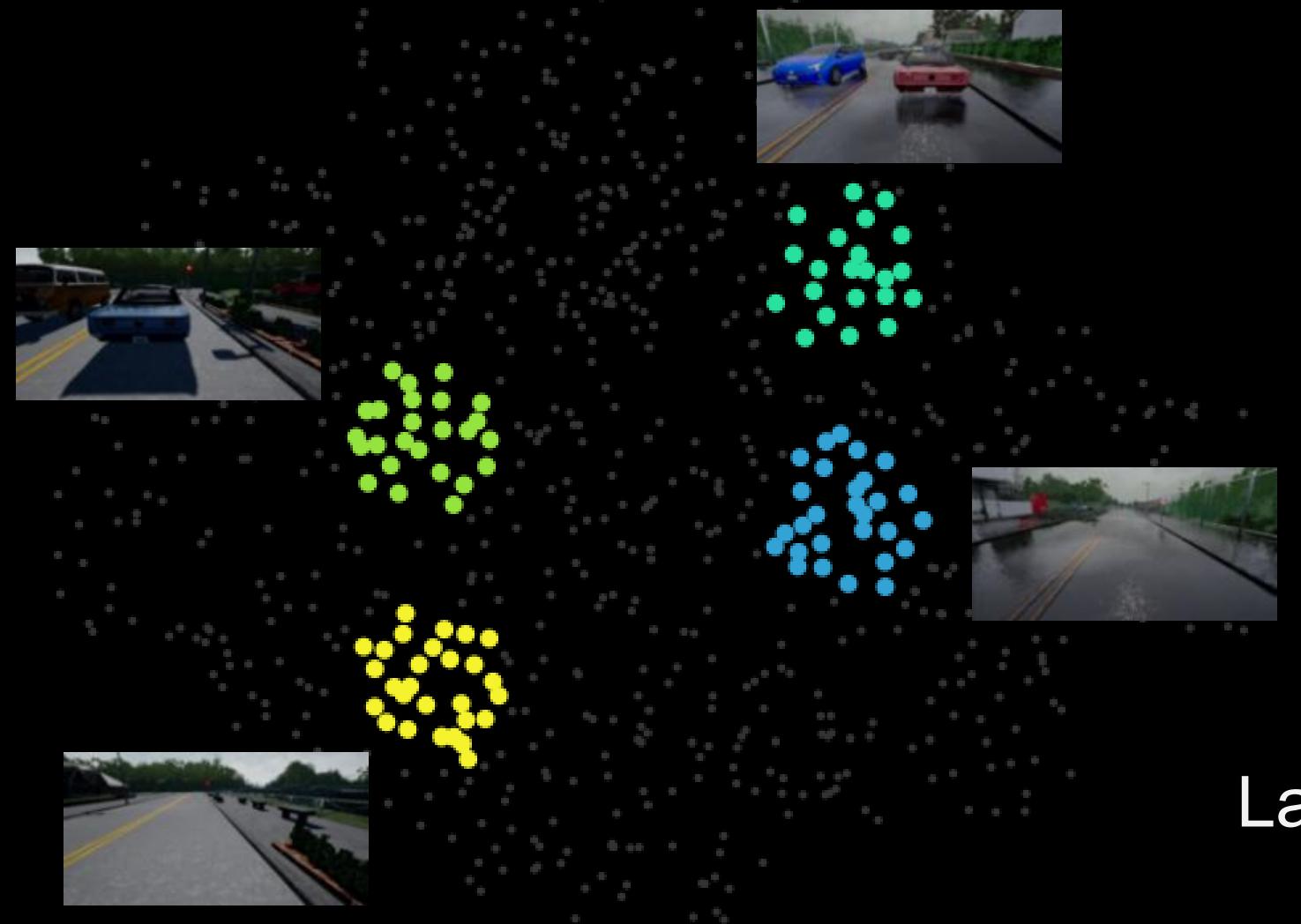
Alec Radford et al. (2016) "Unsupervised representation learning with deep convolutional generative adversarial networks".
4th International Conference on Learning Representations (ICRL) , Conference Track Proceedings.



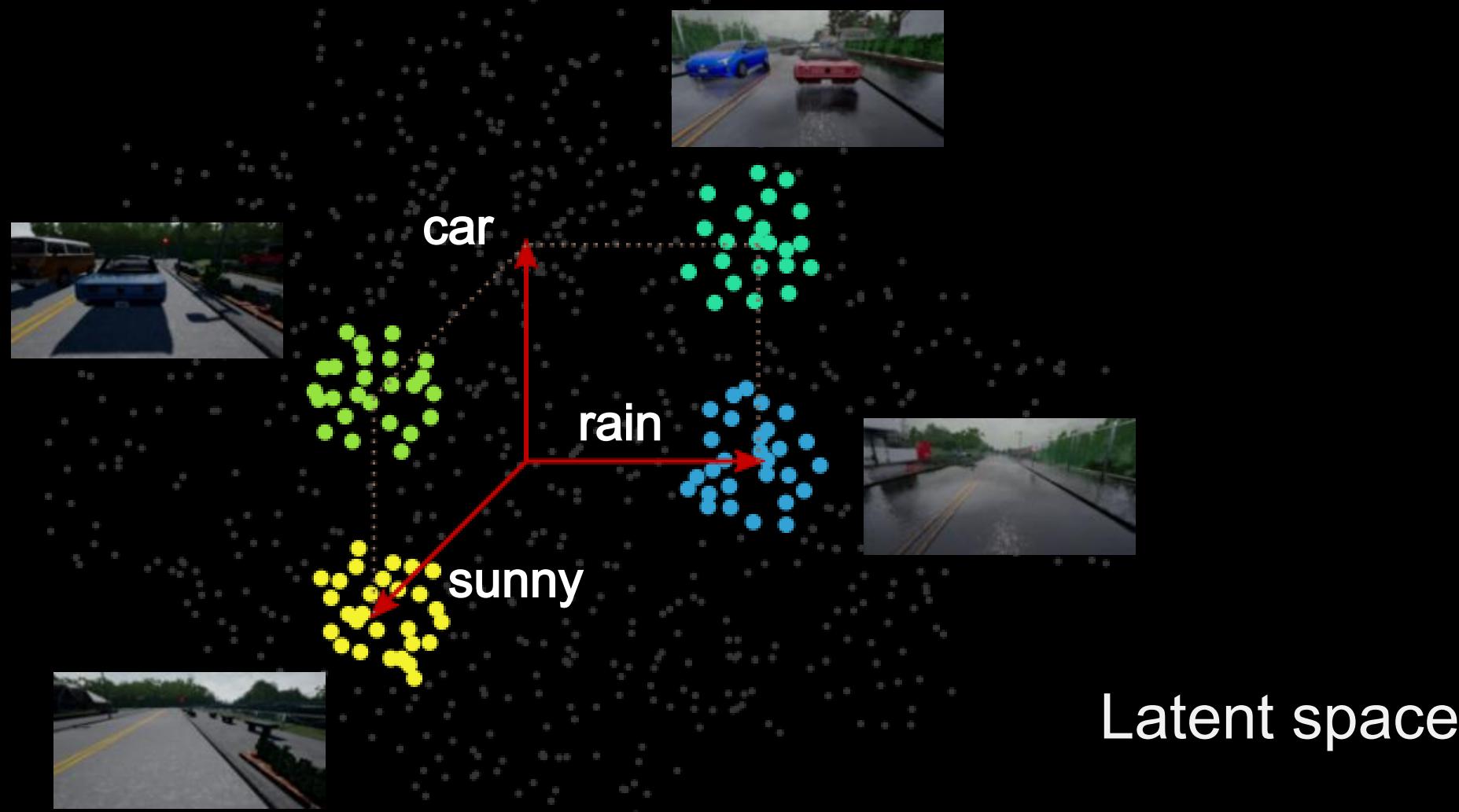
Alec Radford et al. (2016) "Unsupervised representation learning with deep convolutional generative adversarial networks".
4th International Conference on Learning Representations (ICRL) , Conference Track Proceedings.



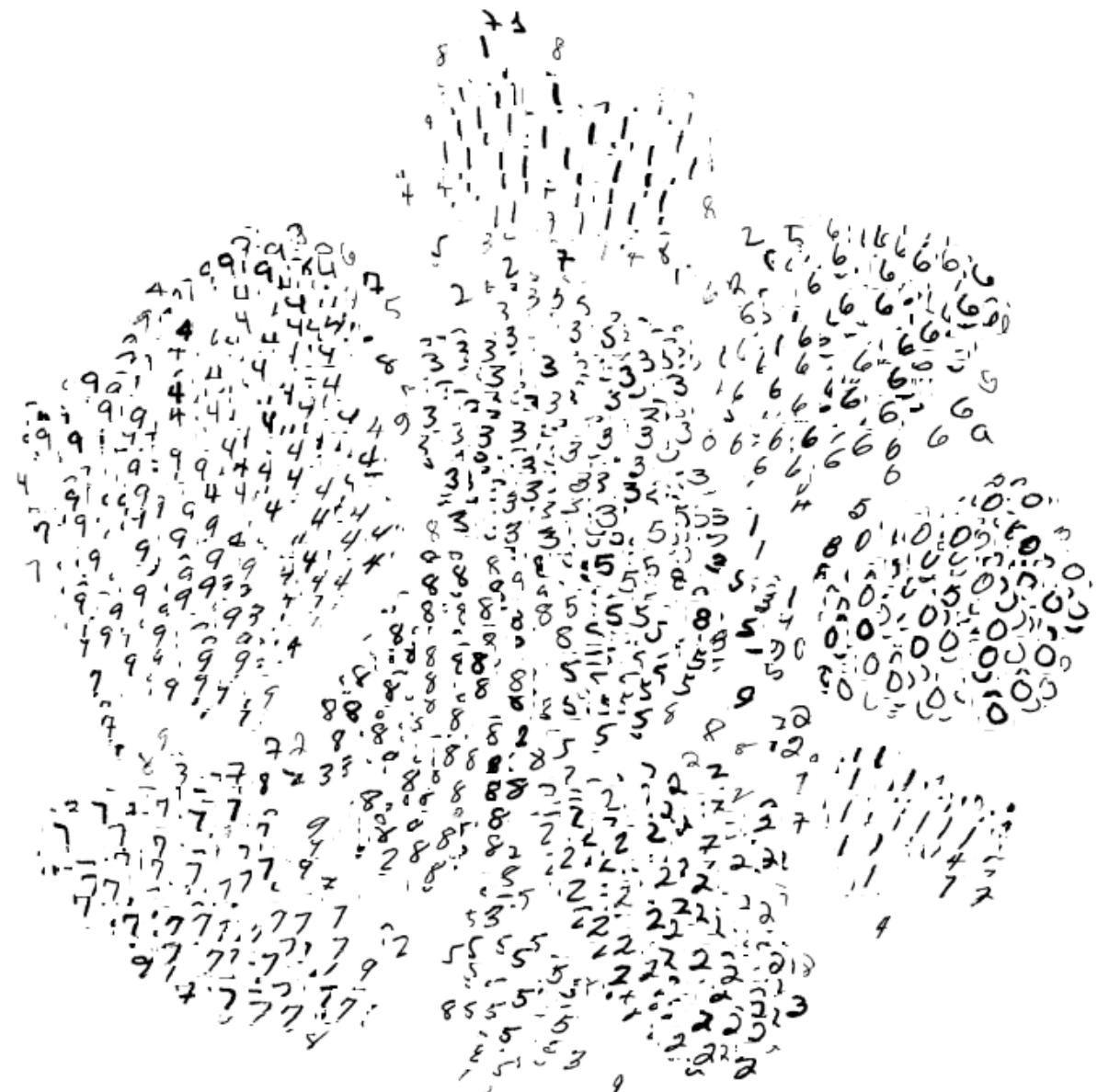
Alec Radford et al. (2016) "Unsupervised representation learning with deep convolutional generative adversarial networks".
4th International Conference on Learning Representations (ICRL) , Conference Track Proceedings.



Latent space



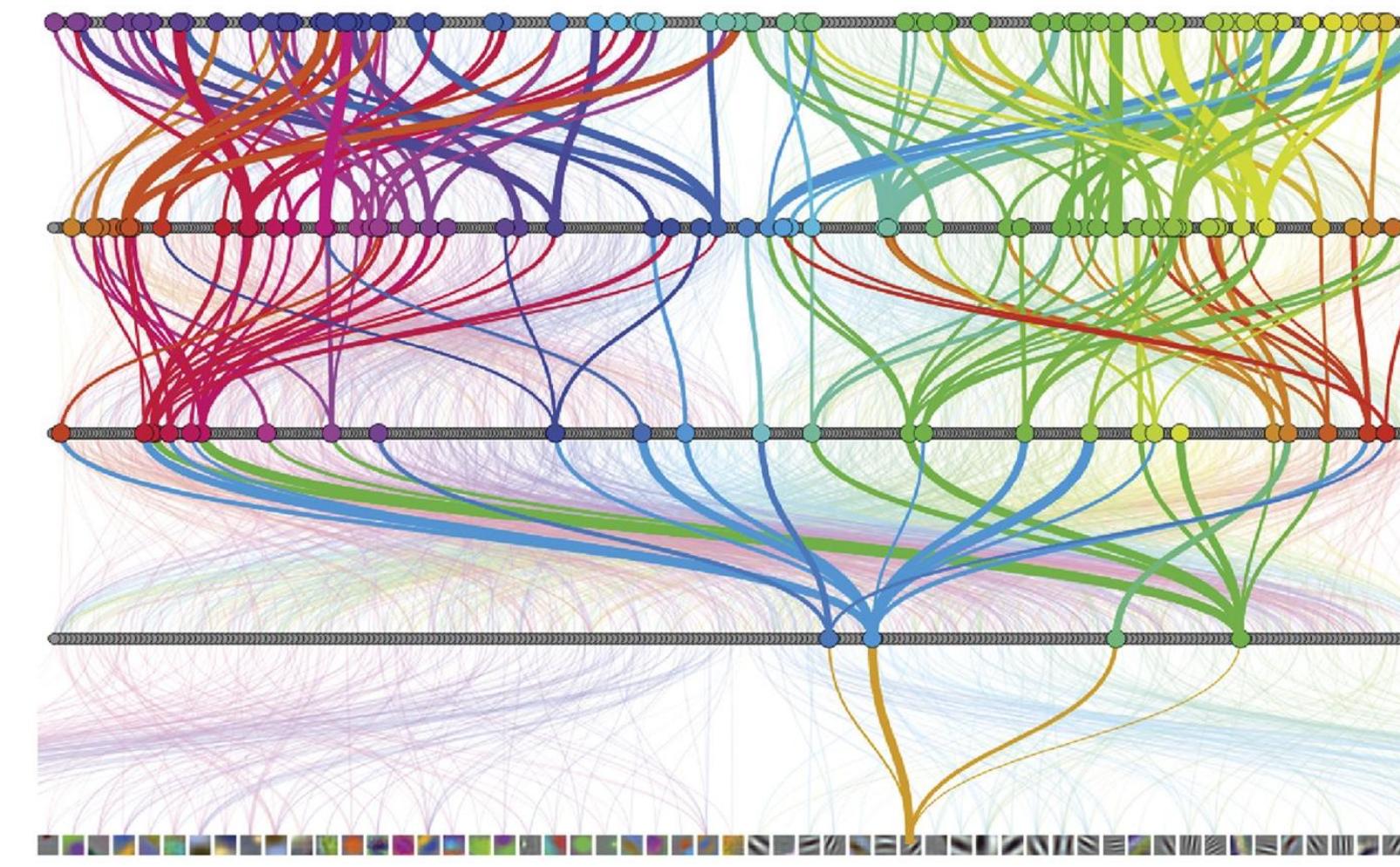
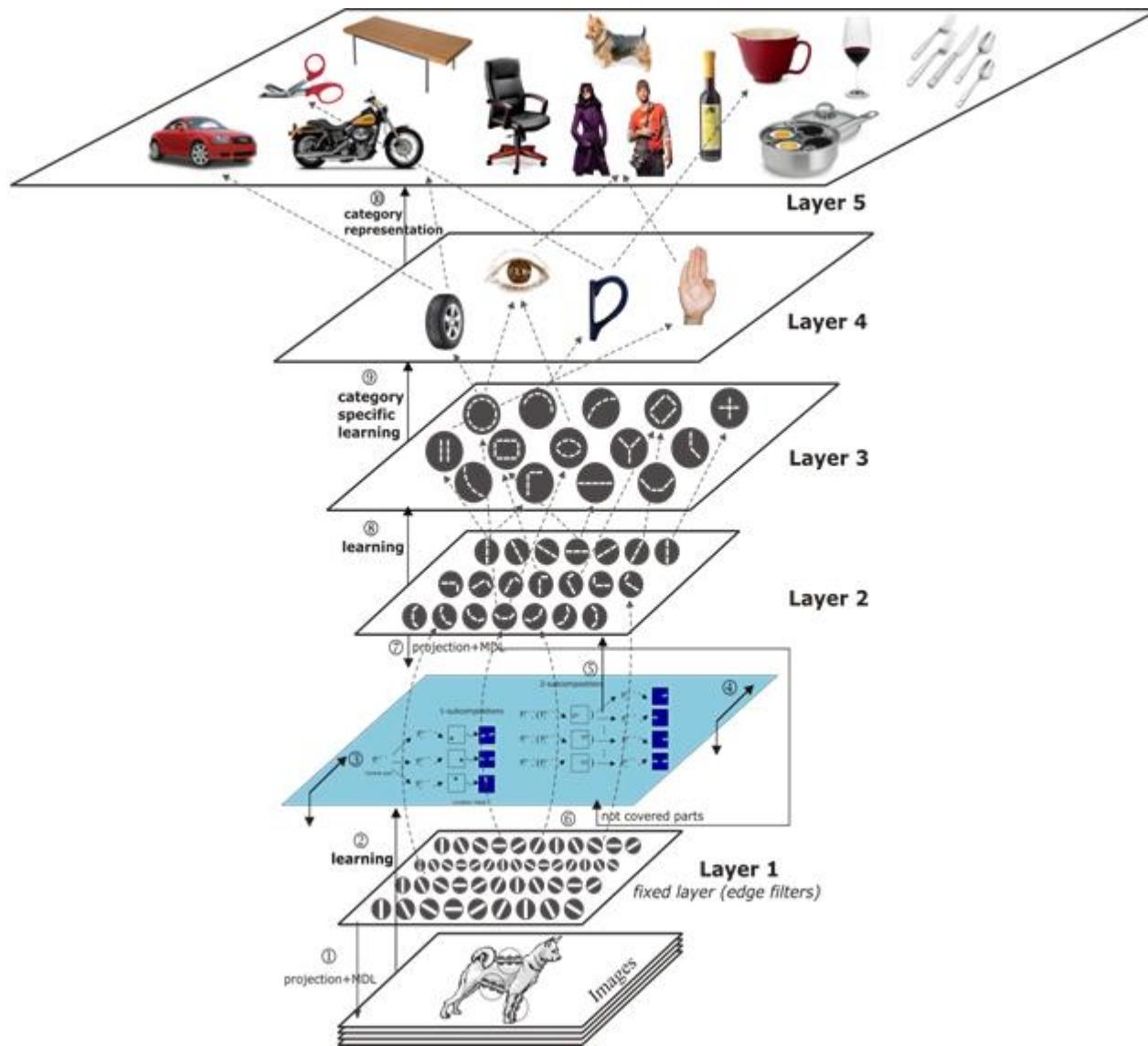
Deep *representations*



<https://projector.tensorflow.org/>

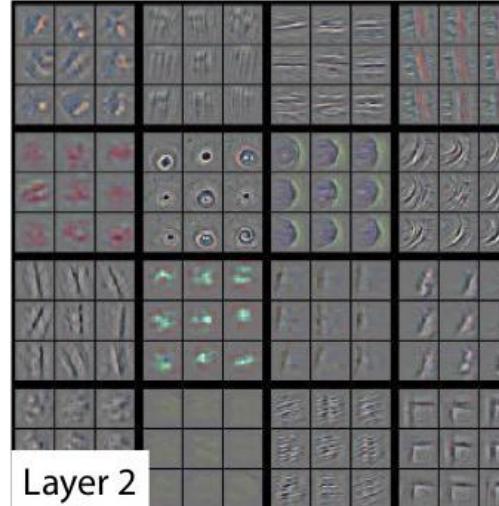


Deep representations



Deep *representations*

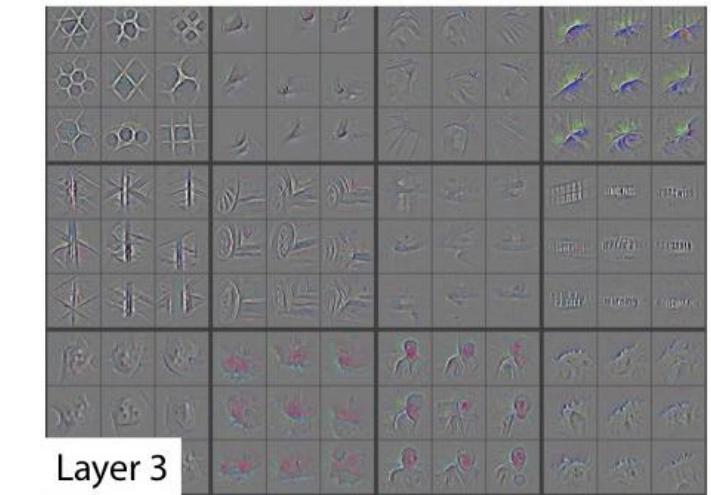
Layer 1



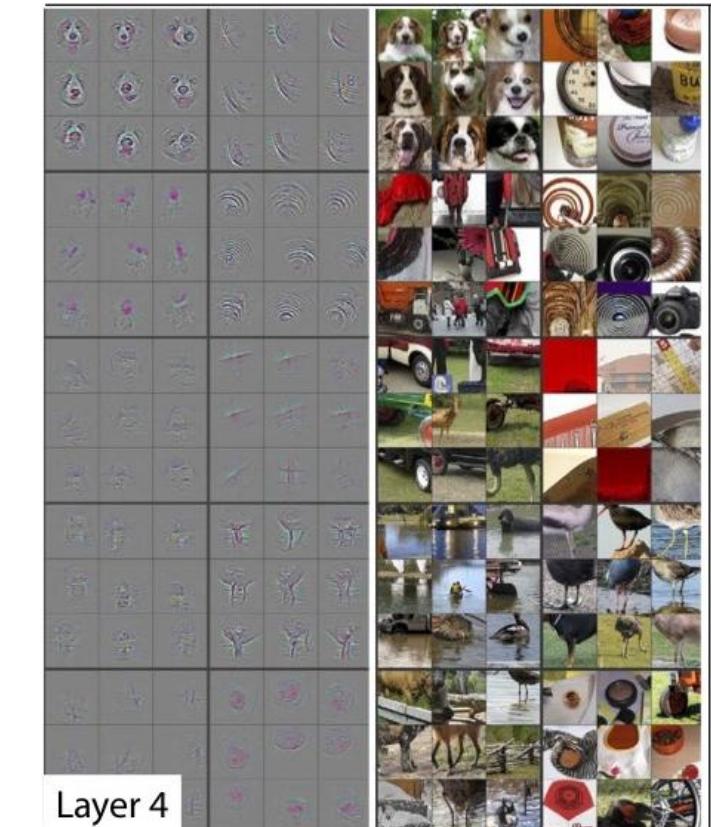
Layer 2



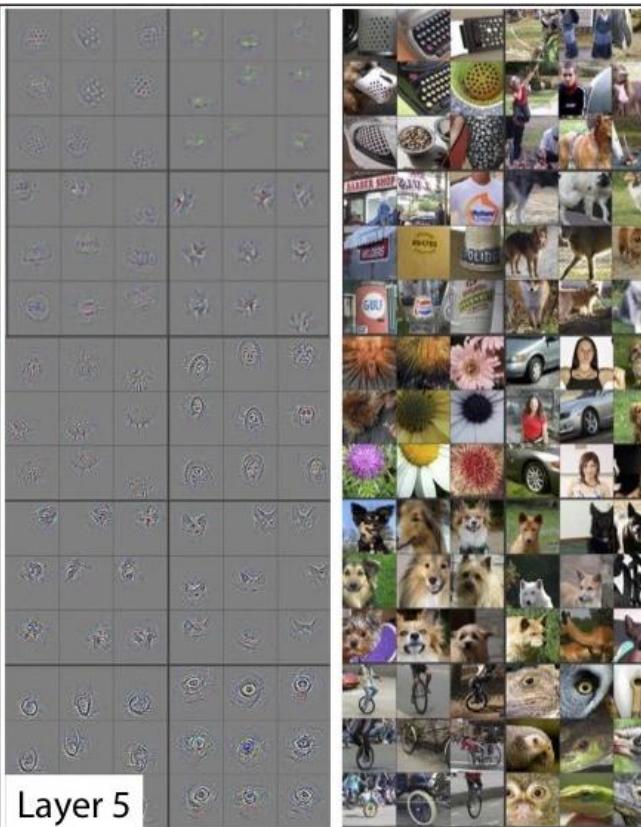
Layer 3



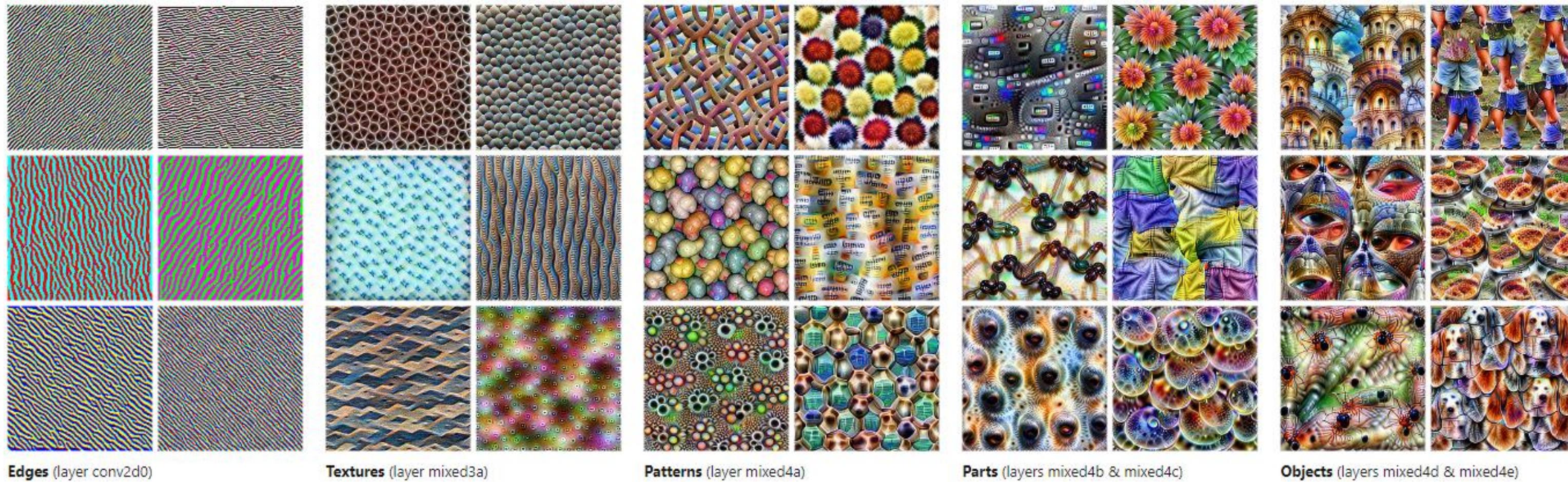
Layer 4



Layer 5



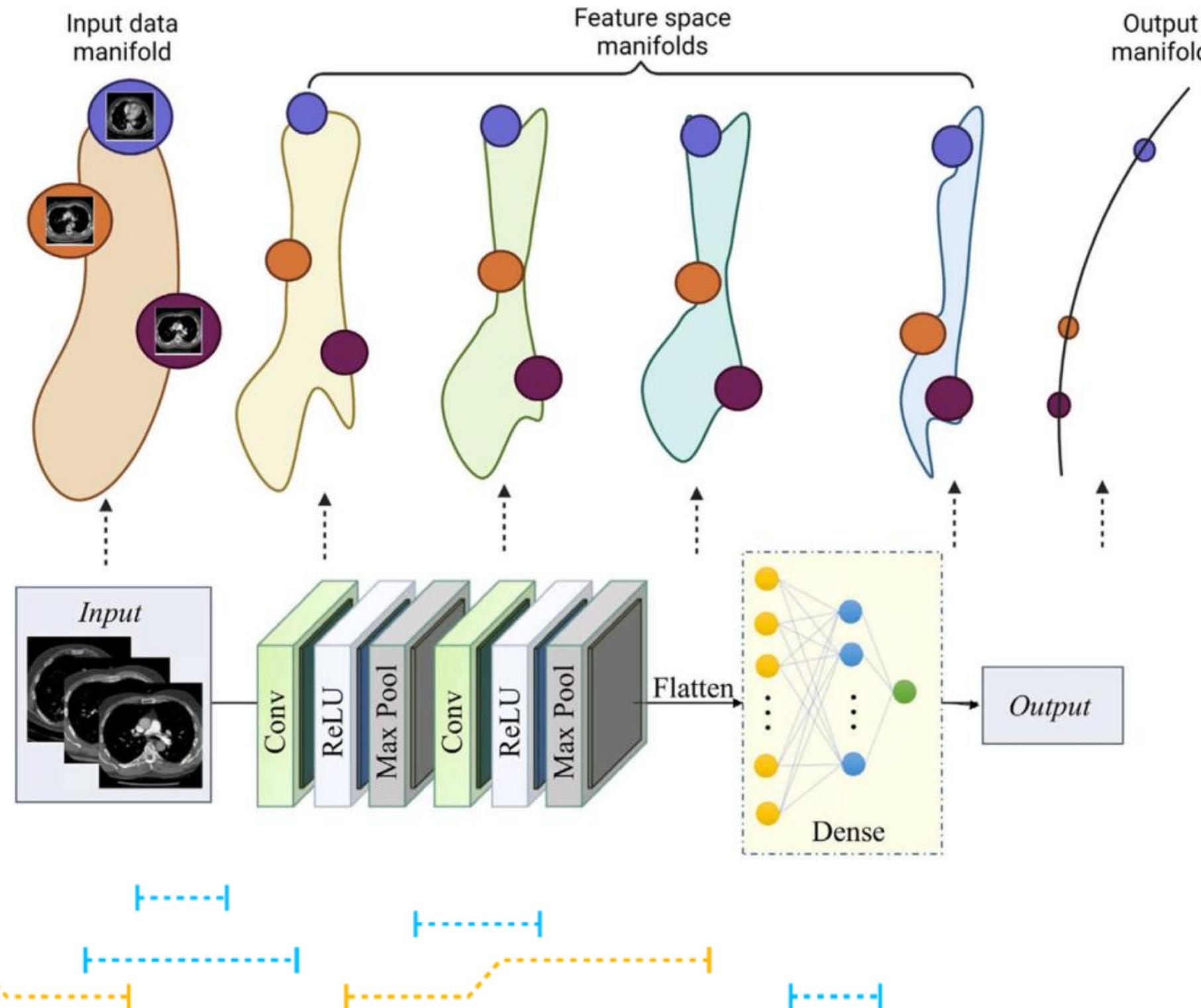
Deep *representations*



<https://distill.pub/2017/feature-visualization/>

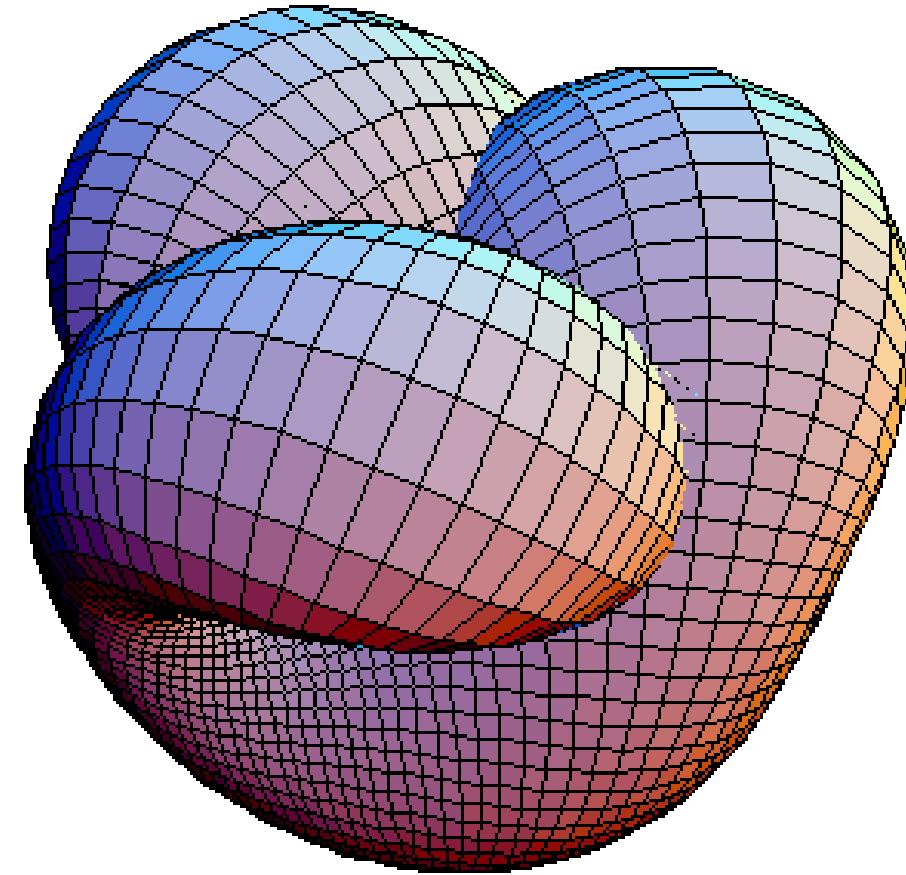


Deep Neural Network *feature space*

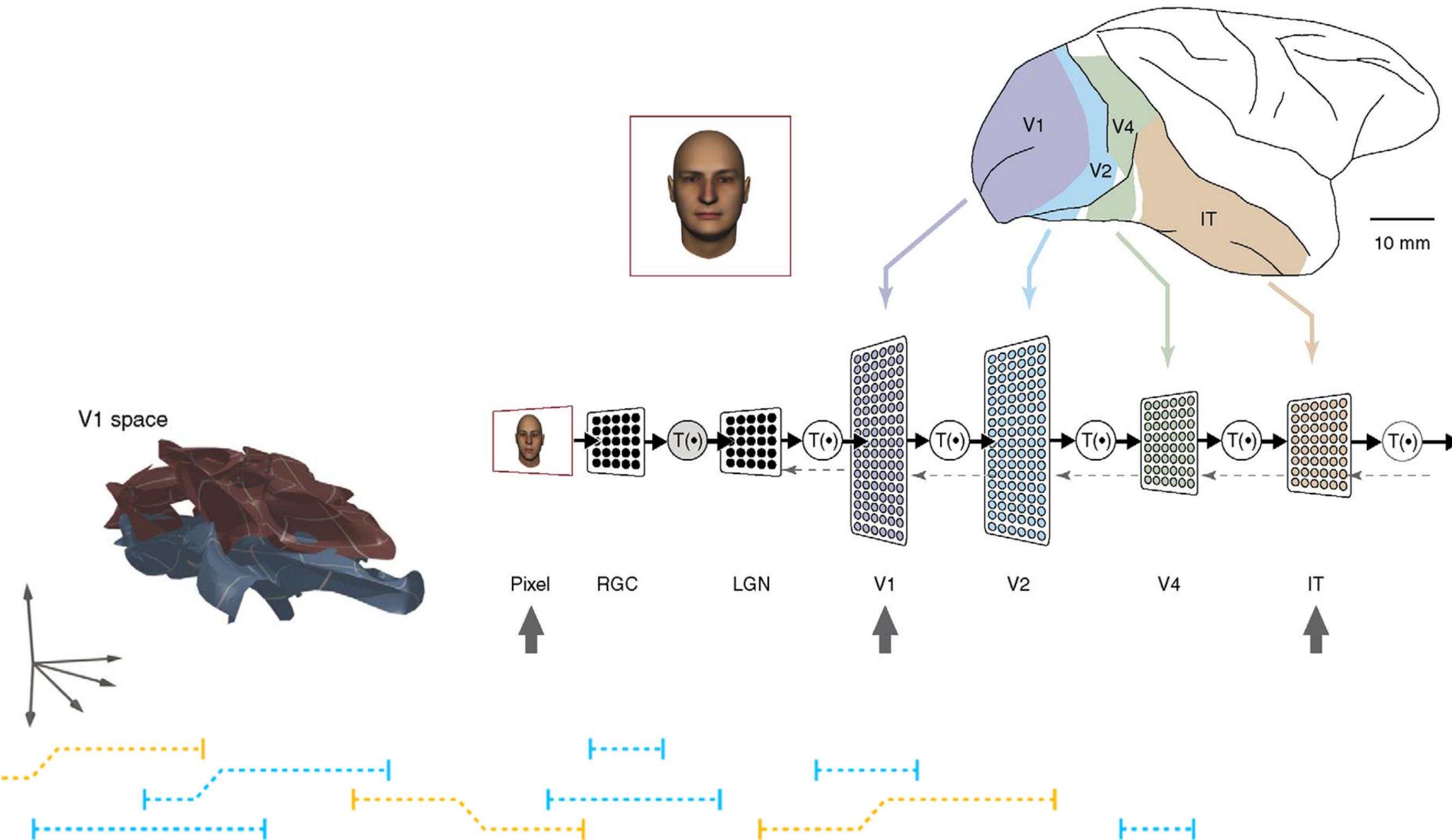


n-manifold

Un *manifold* es un **espacio Hausdorff** segundo contable que es **localmente homeomorfo** a un espacio euclídeo.

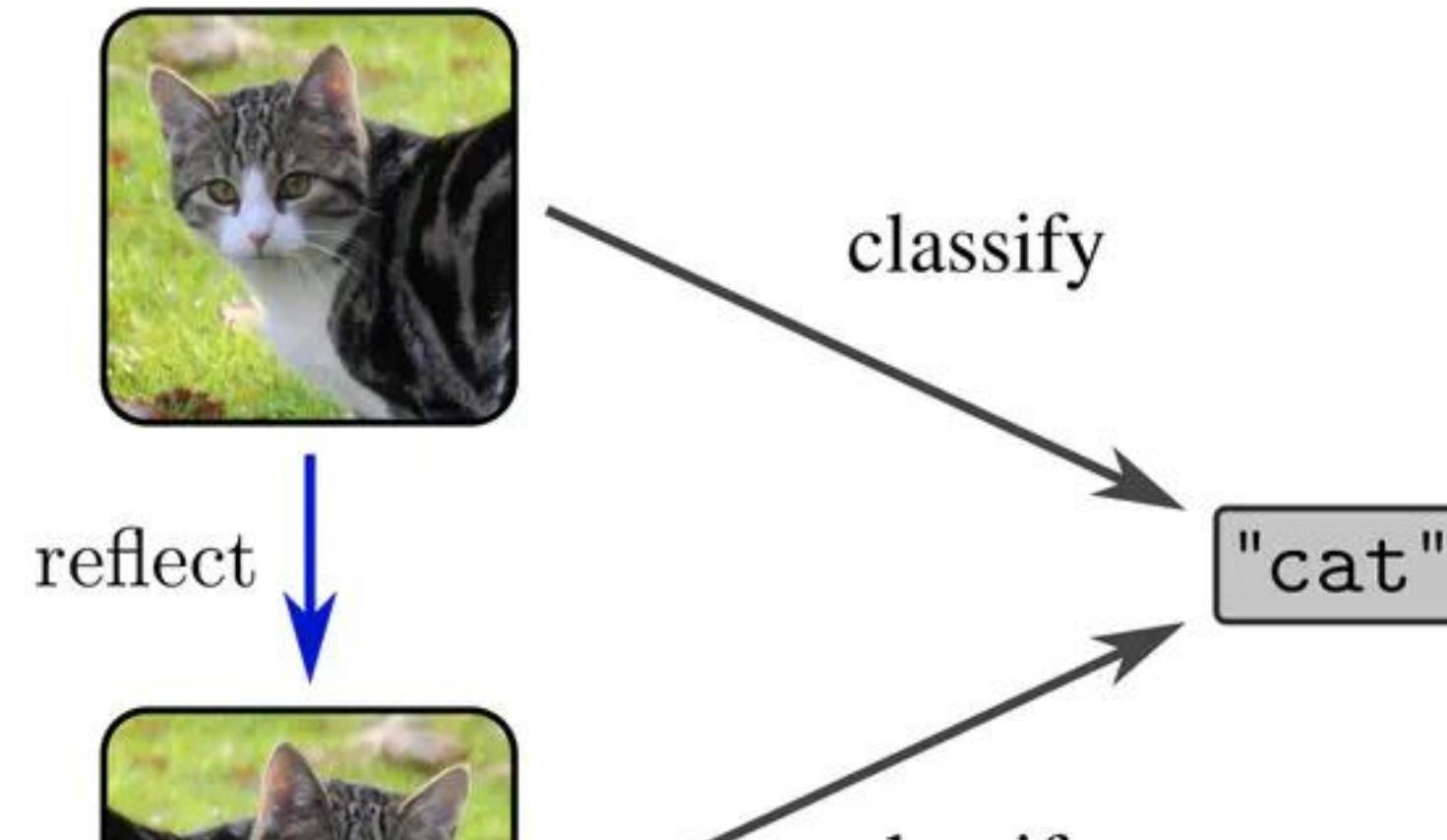


n-manifold



TRANSFORMATEC

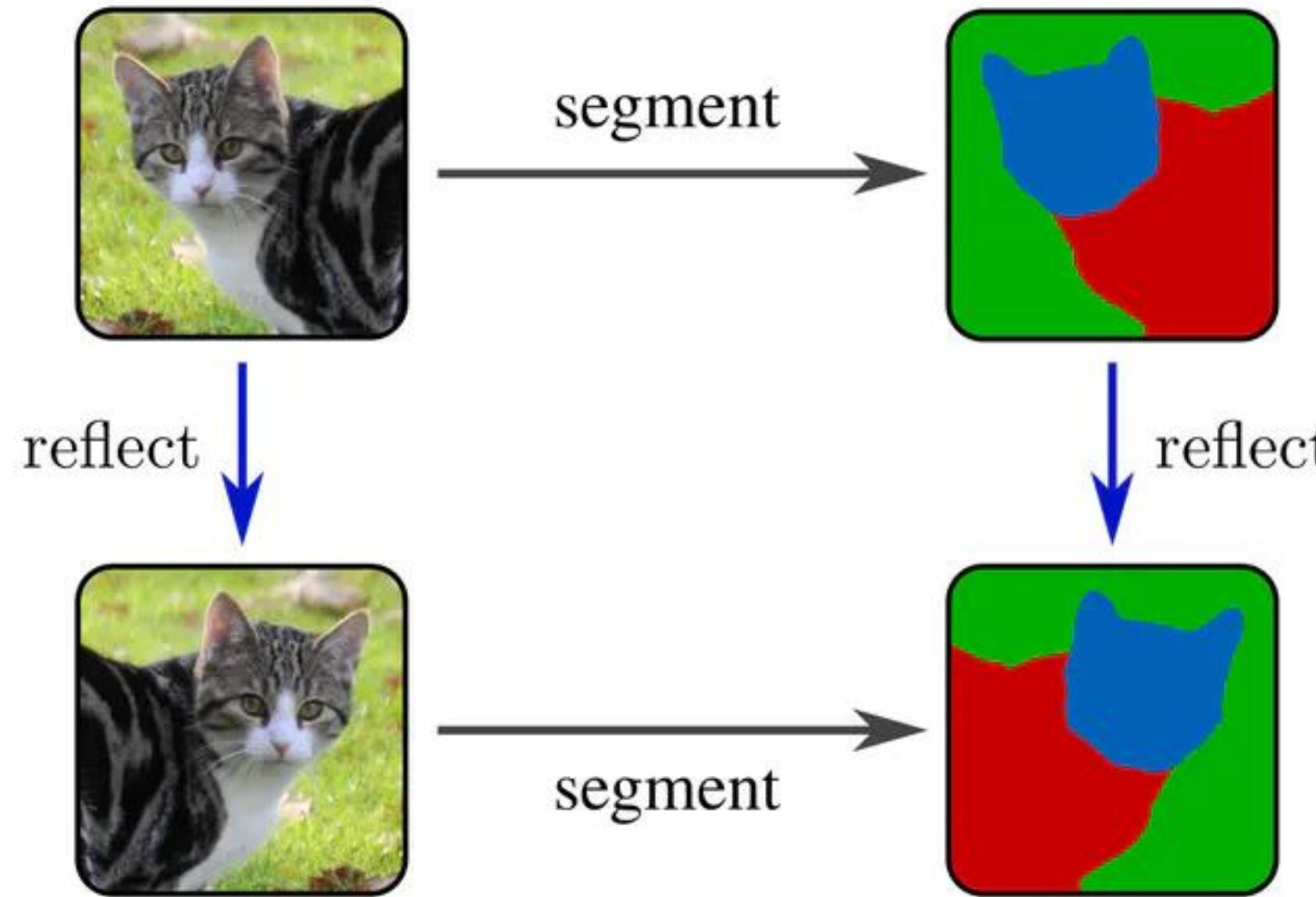
Invariant representation



$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ g \triangleright_X \downarrow & & \downarrow f \\ X & \xrightarrow{f} & Y \end{array}$$



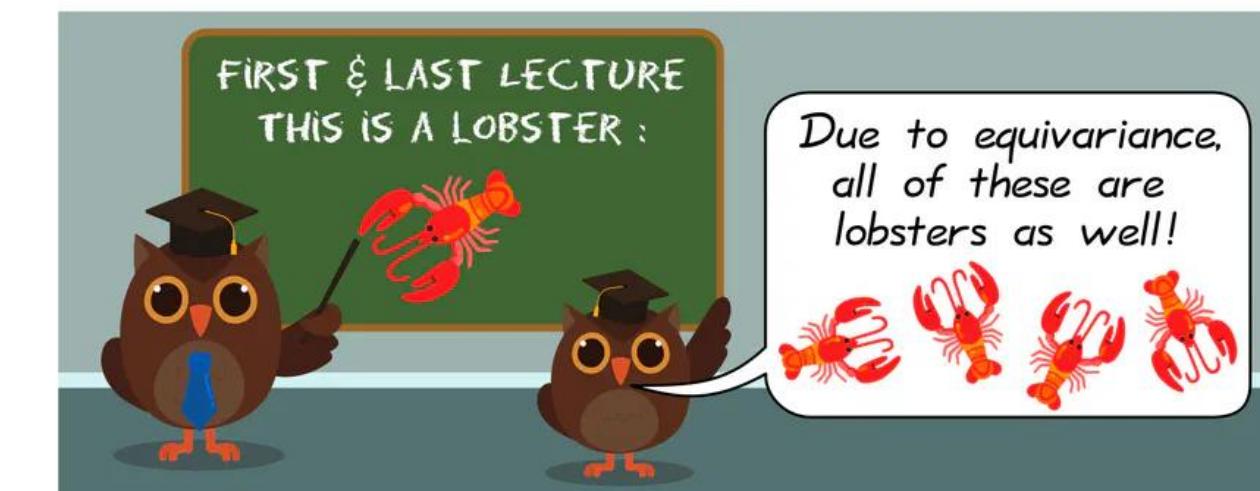
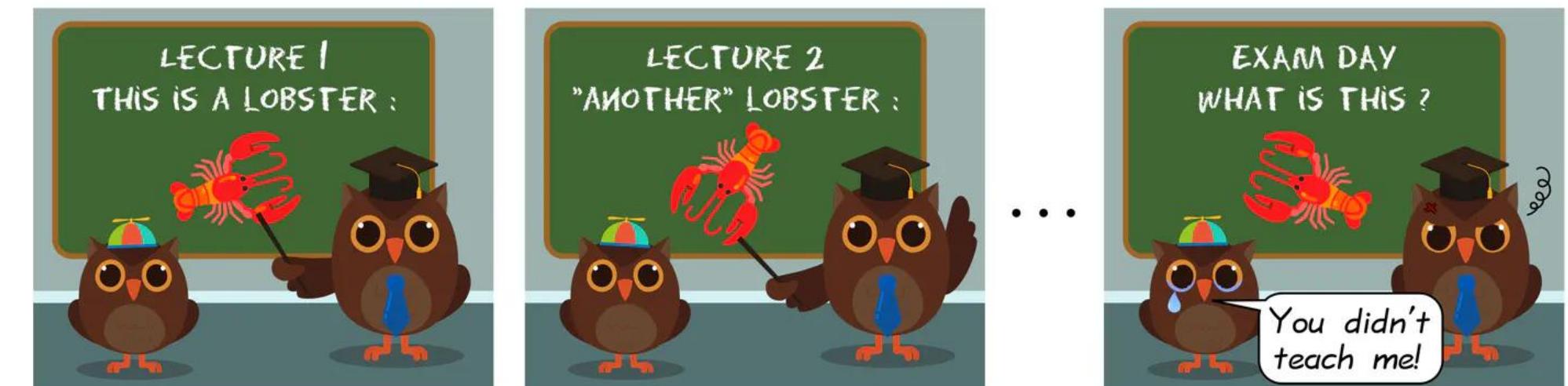
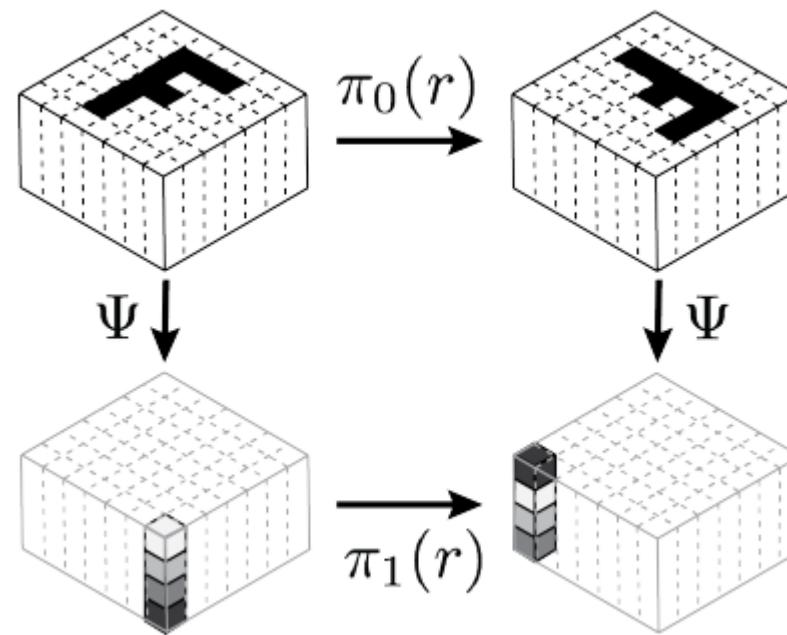
Equivariant *representation*



$$\begin{array}{ccc}
 X & \xrightarrow{f} & Y \\
 g \triangleright_X \downarrow & & \downarrow g \triangleright_Y \\
 X & \xrightarrow{f} & Y
 \end{array}$$

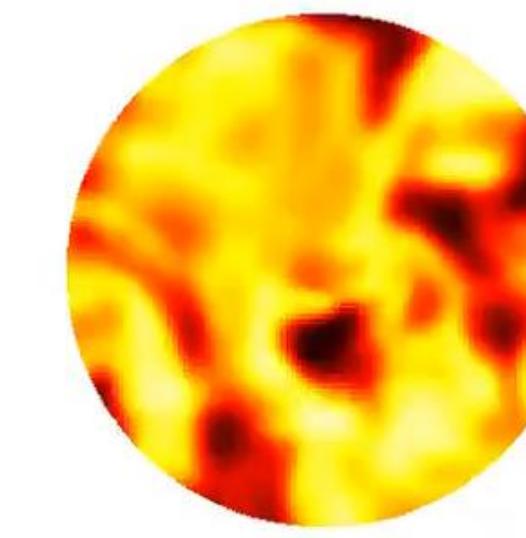
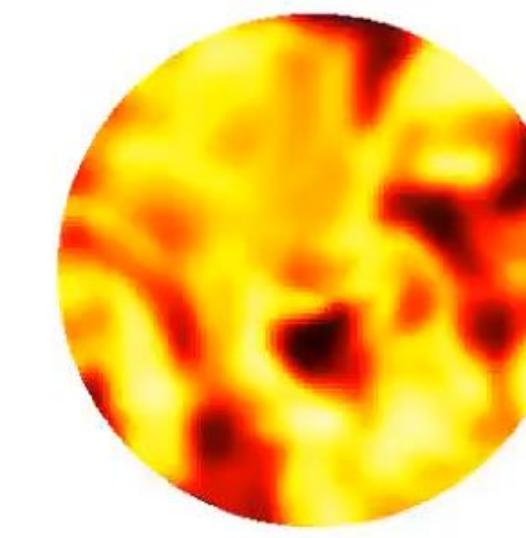


Equivariant representation



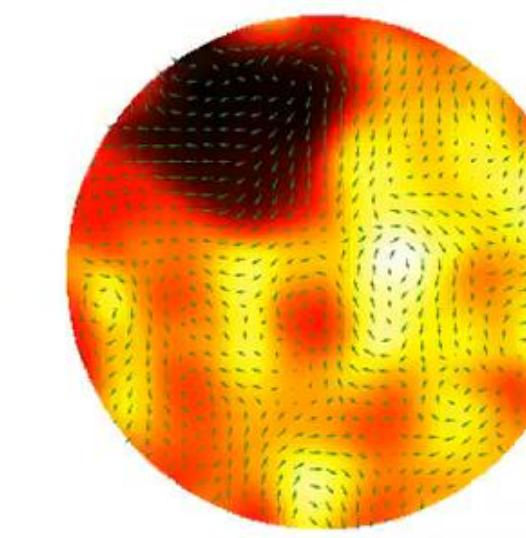
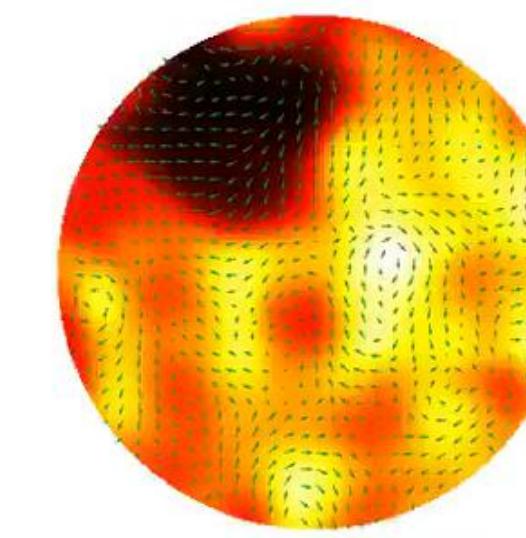
Equivariant *representation*

CNN



La convolución es equivariante de
traslación, pero no de rotación.

G-CNN



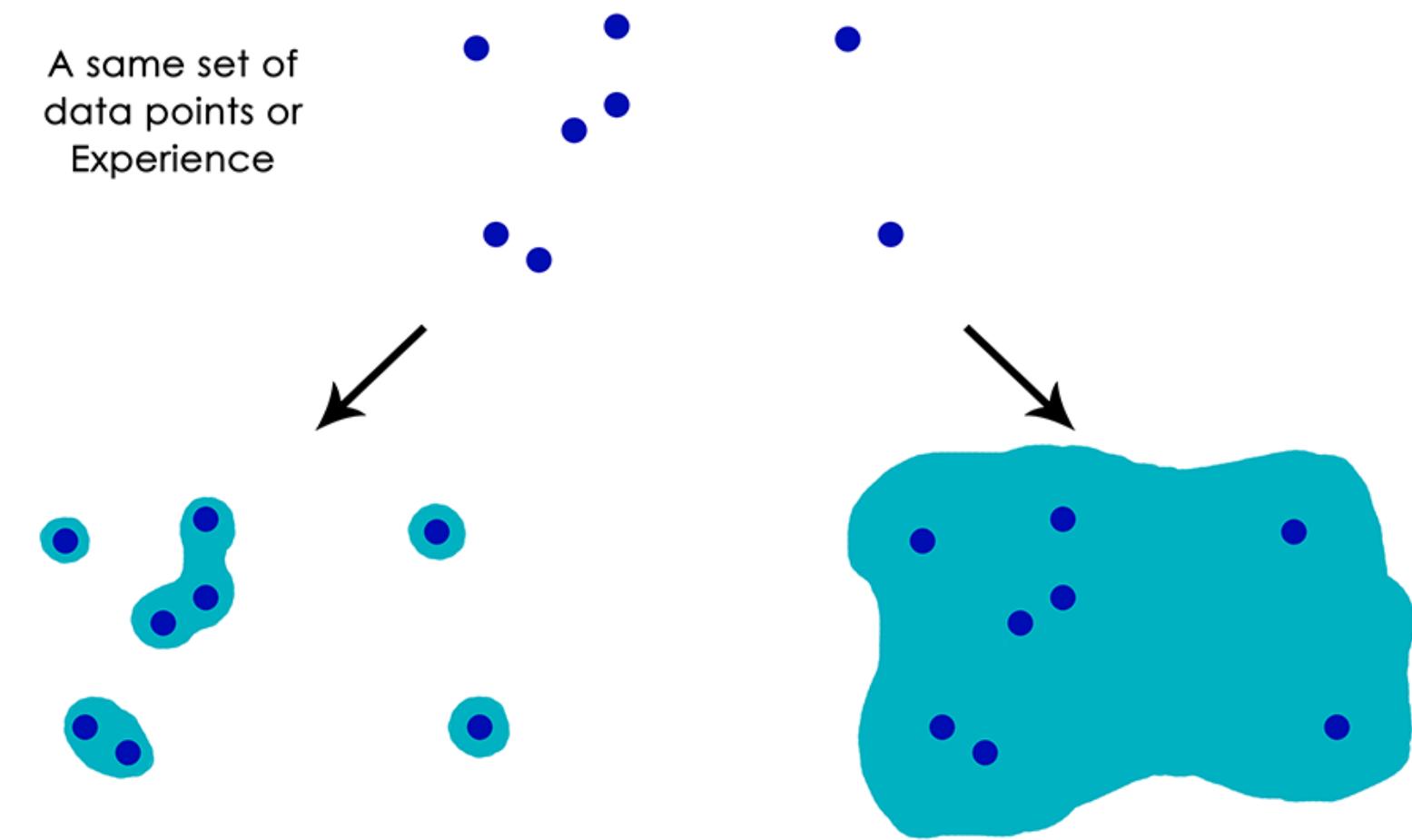
4.



Generalization



Generalization



Local generalization:
Generalization power of
pattern recognition

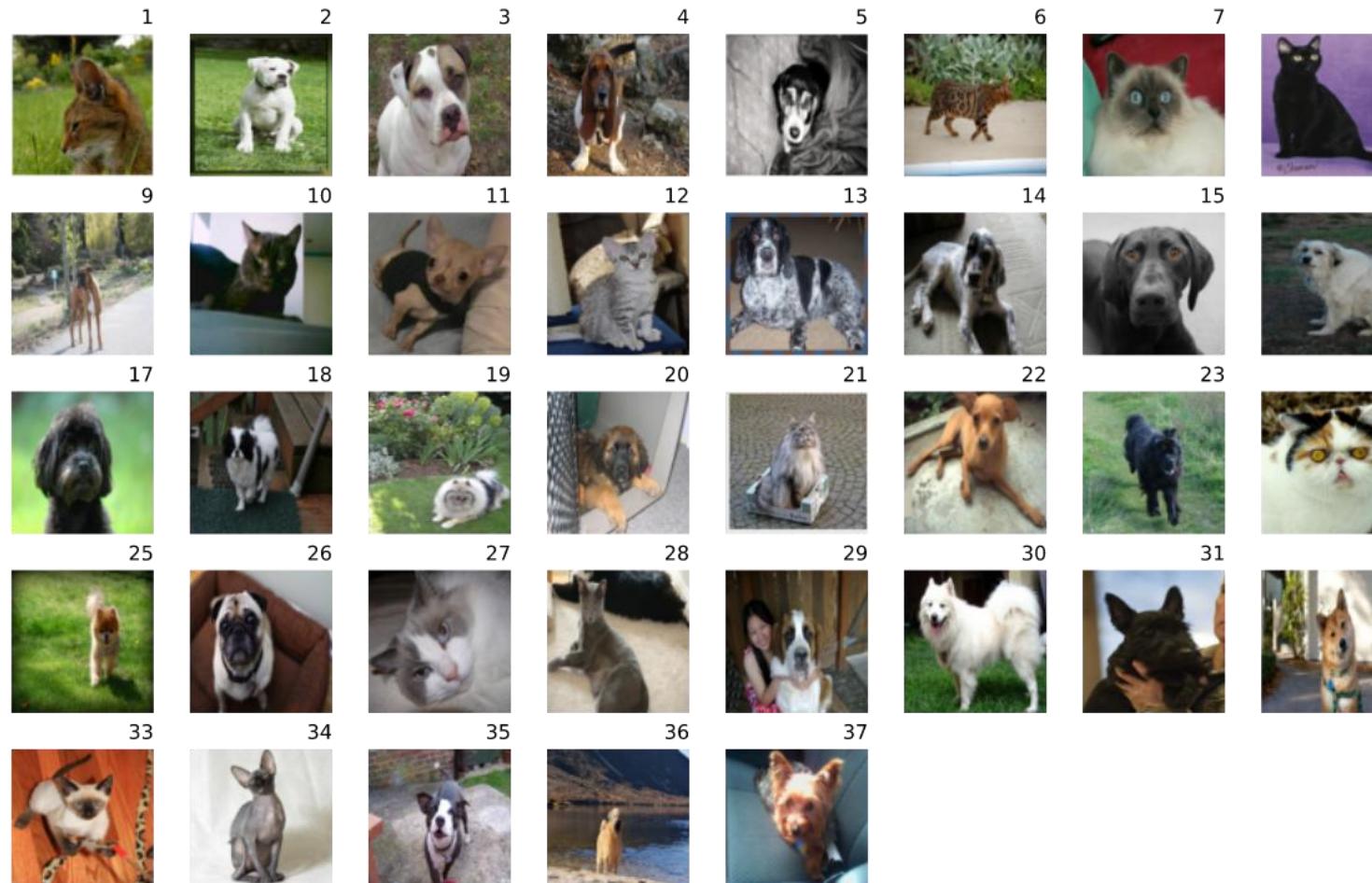
Extreme generalization:
Generalization power
achieved via
abstraction and reasoning



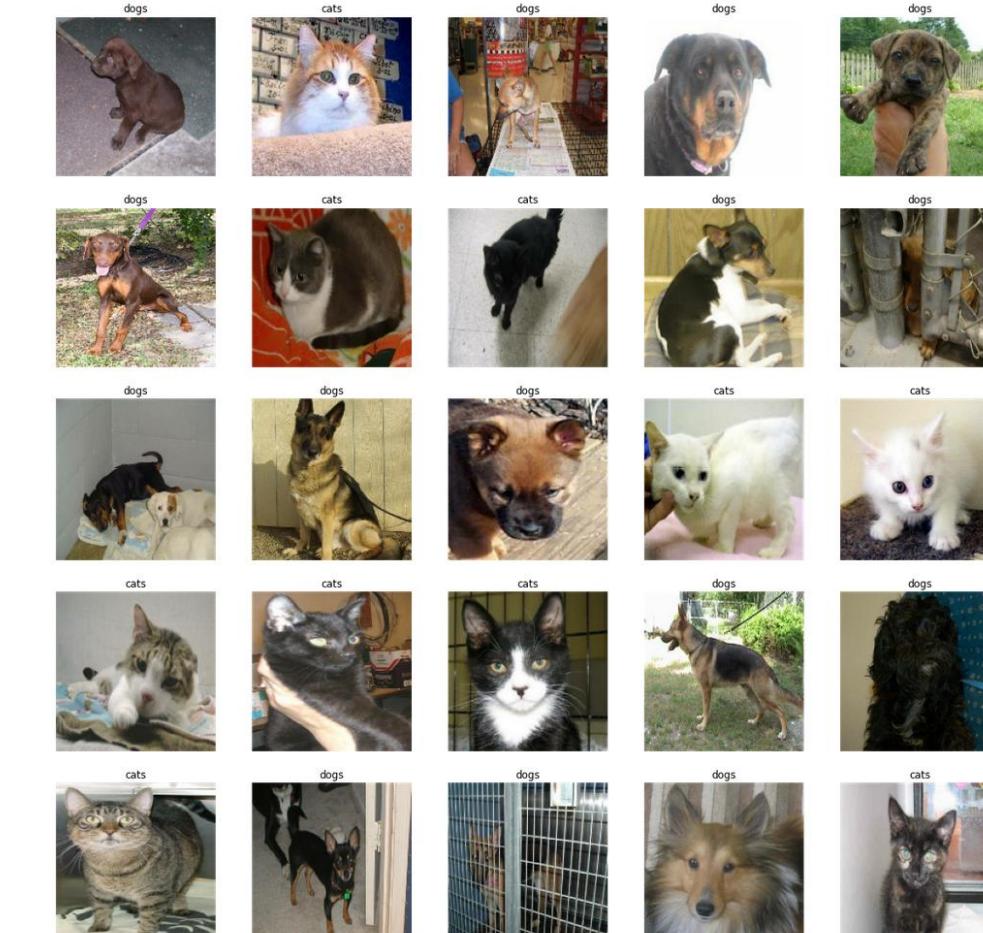
In-Distribution Generalization

(IID Generalization)

El modelo es evaluado en datos que provienen de la misma distribución que los datos de entrenamiento, es decir, se asume que $P_{\text{train}(X,Y)} = P_{\text{test}(X,Y)}$.



Training set



Test set



Intra-Domain *Generalization*

El modelo se evalúa dentro del mismo dominio, pero enfrentando variaciones que no estaban presentes durante el entrenamiento. Estas variaciones pueden incluir cambios en iluminación, orientación, ruido, etc.



Original

Blur

Noise

Brillo

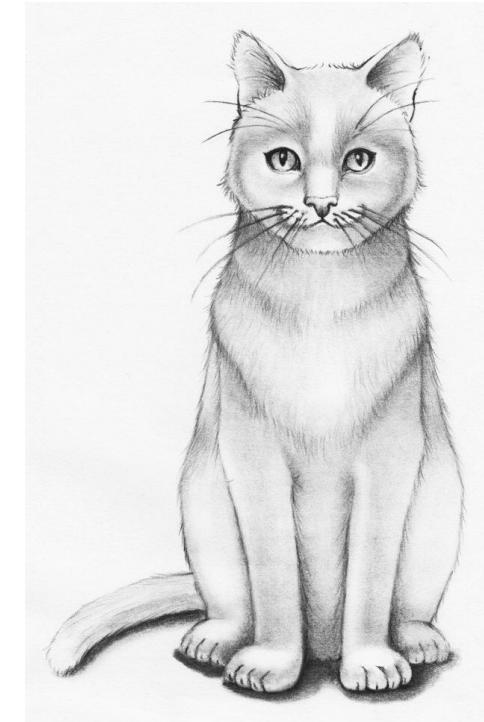
Contraste



Out-of-Distribution Generalization

(OOD Generalization)

El modelo es evaluado con datos que provienen de una distribución diferente a la de entrenamiento, aunque dentro del mismo dominio conceptual



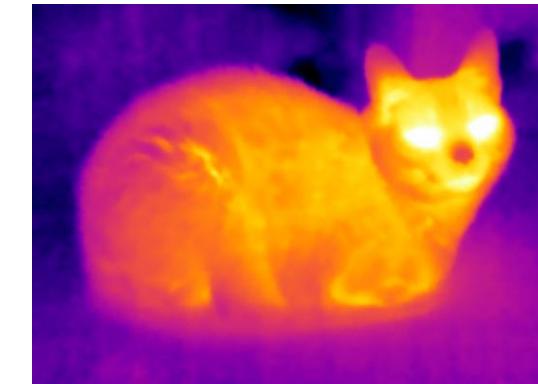
Training set

Test set



Cross-Domain *Generalization*

El modelo es entrenado en un dominio (p. ej., imágenes RGB) y evaluado en un dominio completamente diferente (p. ej., imágenes térmicas).



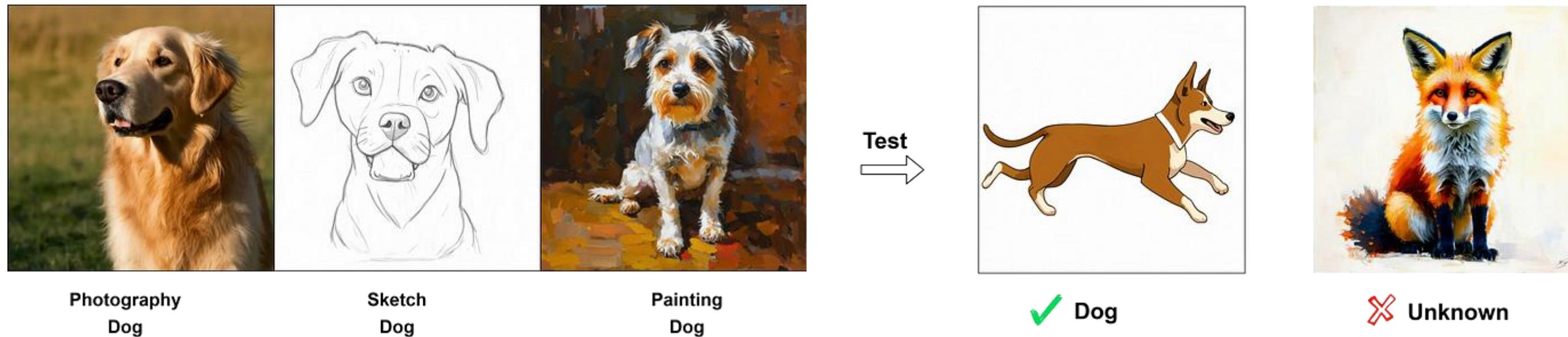
Training set

Test set



Open-Set Domain Generalization

En este caso, el modelo no solo debe enfrentarse a un nuevo dominio, sino que también debe reconocer clases nunca vistas durante el entrenamiento



Task Generalization

El modelo es entrenado para realizar una tarea específica y luego se evalúa en una tarea diferente pero relacionada, sin ser reentrenado



Training



Testing

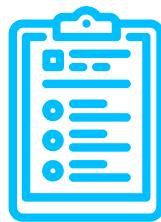


Generalization

	Descripción	Diferencia clave	Ejemplo
In-Distribution	Misma distribución de entrenamiento	No hay cambios en la distribución	Entrenar y probar en imágenes de gatos similares
Intra-Domain	Mismo dominio con ligeras variaciones	Cambios menores dentro del dominio	Diferentes condiciones de iluminación
Out-of-Distribution (OOD)	Nueva distribución pero mismo dominio	Datos fuera de la distribución de entrenamiento	Dibujos, modelos 3D de gatos
Cross-Domain	Dominio completamente diferente	Cambio en la naturaleza de los datos	Imágenes térmicas de gatos
Open-Set Domain	Nuevo dominio + nuevas clases desconocidas	Nuevas clases en un entorno diferente	Clasificar nuevos animales nunca vistos
Task Generalization	Aprender en una tarea y generalizar a otra	Cambio de tarea	..



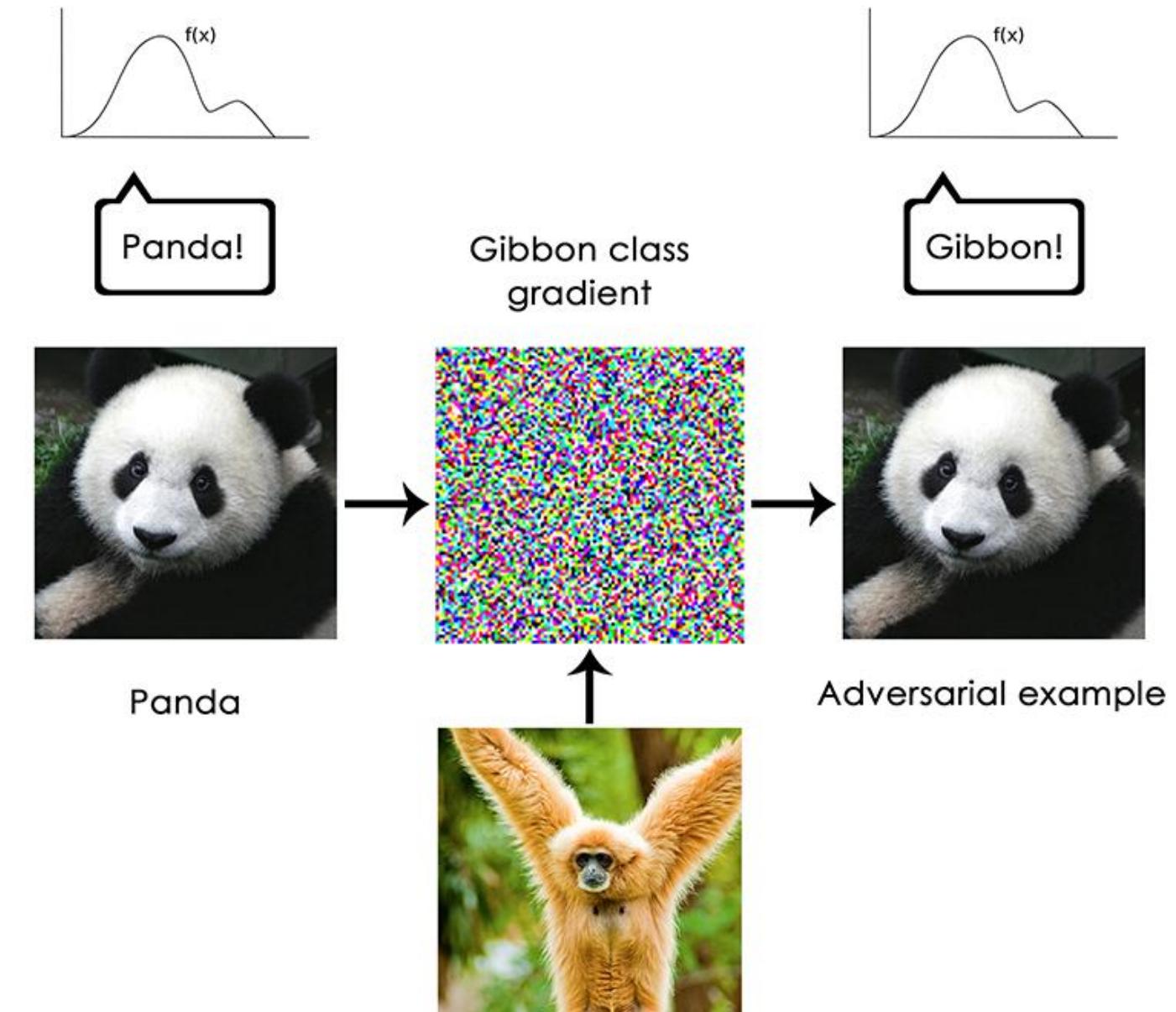
5.



Adversarial *Attack*



Adversarial Attack



Adversarial Attack

White-box attacks

El atacante tiene acceso completo al modelo, incluyendo su arquitectura, parámetros y datos de entrenamiento. Esto permite generar ataques muy precisos y efectivos.

Fast Gradient Sign Method (FGSM)

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y))$$

Donde:

- x entrada original (ej. imagen)
- y etiqueta verdadera de la entrada
- $J(x, y)$ función de pérdida del modelo
- $\nabla_x J(x, y)$ gradiente de la pérdida con respecto a la entrada x
- $\text{sign}(\cdot)$ función signo, que devuelve +1 o -1 por cada componente del gradiente
- ϵ parámetro de perturbación (determina la magnitud de la perturbación)



Adversarial Attack

White-box attacks

El atacante tiene acceso completo al modelo, incluyendo su arquitectura, parámetros y datos de entrenamiento. Esto permite generar ataques muy precisos y efectivos.

Fast Gradient Sign Method (FGSM)

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y))$$

Donde:

- x entrada original (ej. imagen)
- y etiqueta verdadera de la entrada
- $J(x, y)$ función de pérdida del modelo
- $\nabla_x J(x, y)$ gradiente de la pérdida con respecto a la entrada x
- $\text{sign}(\cdot)$ función signo, que devuelve +1 o -1 por cada componente del gradiente
- ϵ parámetro de perturbación (determina la magnitud de la perturbación)

Projected Gradient Descent (PGD)

$$x_{t+1} = \text{Proj}_{B_\epsilon(x)} \left(x_t + \alpha \cdot \text{sign}(\nabla_x J(x, y)) \right)$$

Donde:

- x_t entrada adversarial en la iteración actual
- α paso de actualización en cada iteración (usualmente $\alpha < \epsilon$)
- $\text{sign}(\nabla_x J(x, y))$ dirección del gradiente de la pérdida respecto a la entrada
- $\text{Proj}_{B_\epsilon(x)}$ proyección en una ball L_∞ de radio ϵ centrada en x , asegurando que la perturbación no supere el límite establecido



Adversarial Attack

Black-box attacks

El atacante no tiene acceso al modelo, solo puede observar las respuestas del modelo a diferentes entradas.

Algorithm 1 SimBA in Pseudocode

```

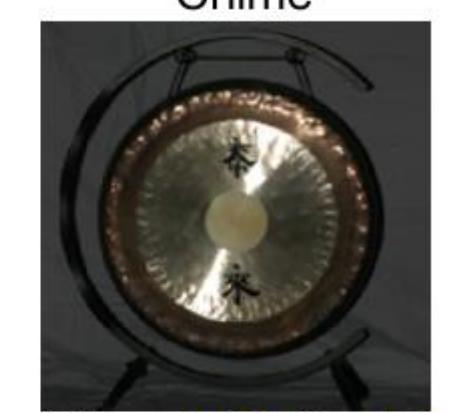
1: procedure SIMBA( $\mathbf{x}, y, Q, \epsilon$ )
2:    $\delta = \mathbf{0}$ 
3:    $\mathbf{p} = p_h(y | \mathbf{x})$ 
4:   while  $\mathbf{p}_y = \max_{y'} \mathbf{p}_{y'}$  do
5:     Pick randomly without replacement:  $\mathbf{q} \in Q$ 
6:     for  $\alpha \in \{\epsilon, -\epsilon\}$  do
7:        $\mathbf{p}' = p_h(y | \mathbf{x} + \delta + \alpha\mathbf{q})$ 
8:       if  $\mathbf{p}'_y < \mathbf{p}_y$  then
9:          $\delta = \delta + \alpha\mathbf{q}$ 
10:         $\mathbf{p} = \mathbf{p}'$ 
11:      break
return  $\delta$ 
```



Original



SimBA



Gong



Maltese dog



Chime



Lhasa

$\|\delta\|_2 = 0.97, B = 76$

TRANSFORMATEC

Chuan Guo et al. (2019) "Simple Black-box Adversarial Attacks"
International conference on machine learning (pp. 2484-2493). PMLR.

GRACIAS

Victor Flores Benites