

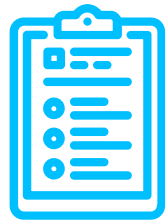
# Sesión 5.2

## *Teacher-student Architecture*

*DeiT. Noisy Student.*

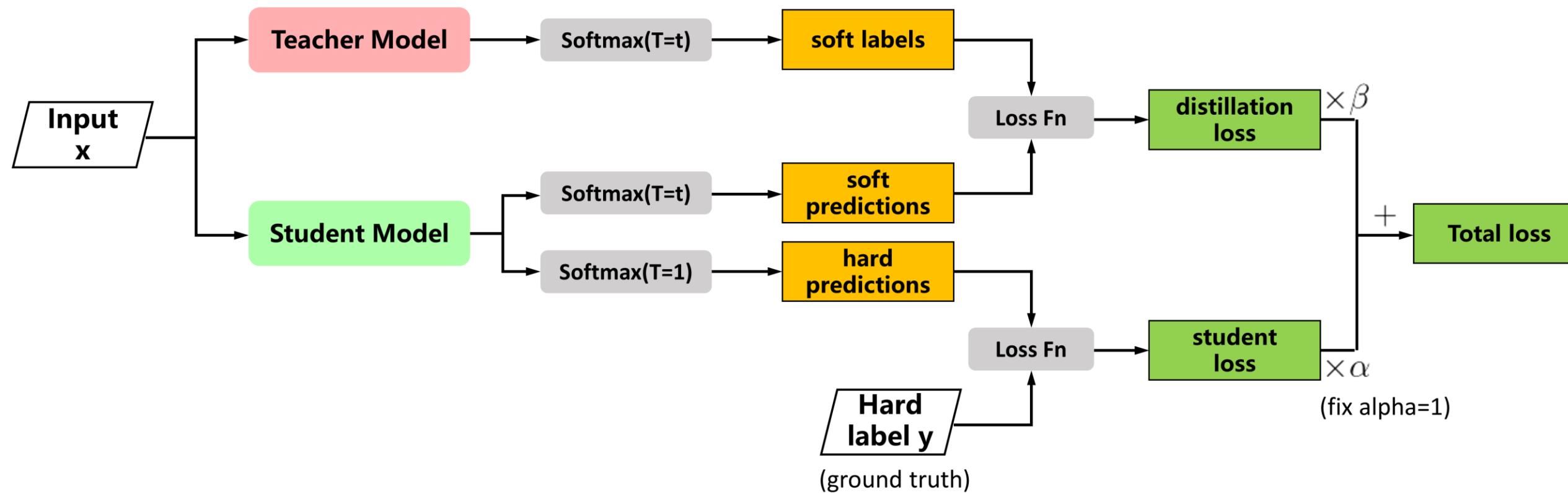


1.



## **Knowledge** *distillation*

# Knowledge *distillation*



```
kd = KnowledgeDistill(kd_teacher_model, kd_T)
```

kd\_teacher\_model: **Teacher Model**

kd\_T: t

```
loss = kd.loss(data, student_out)
```

data: **Input x**

student\_out: **hard predictions**



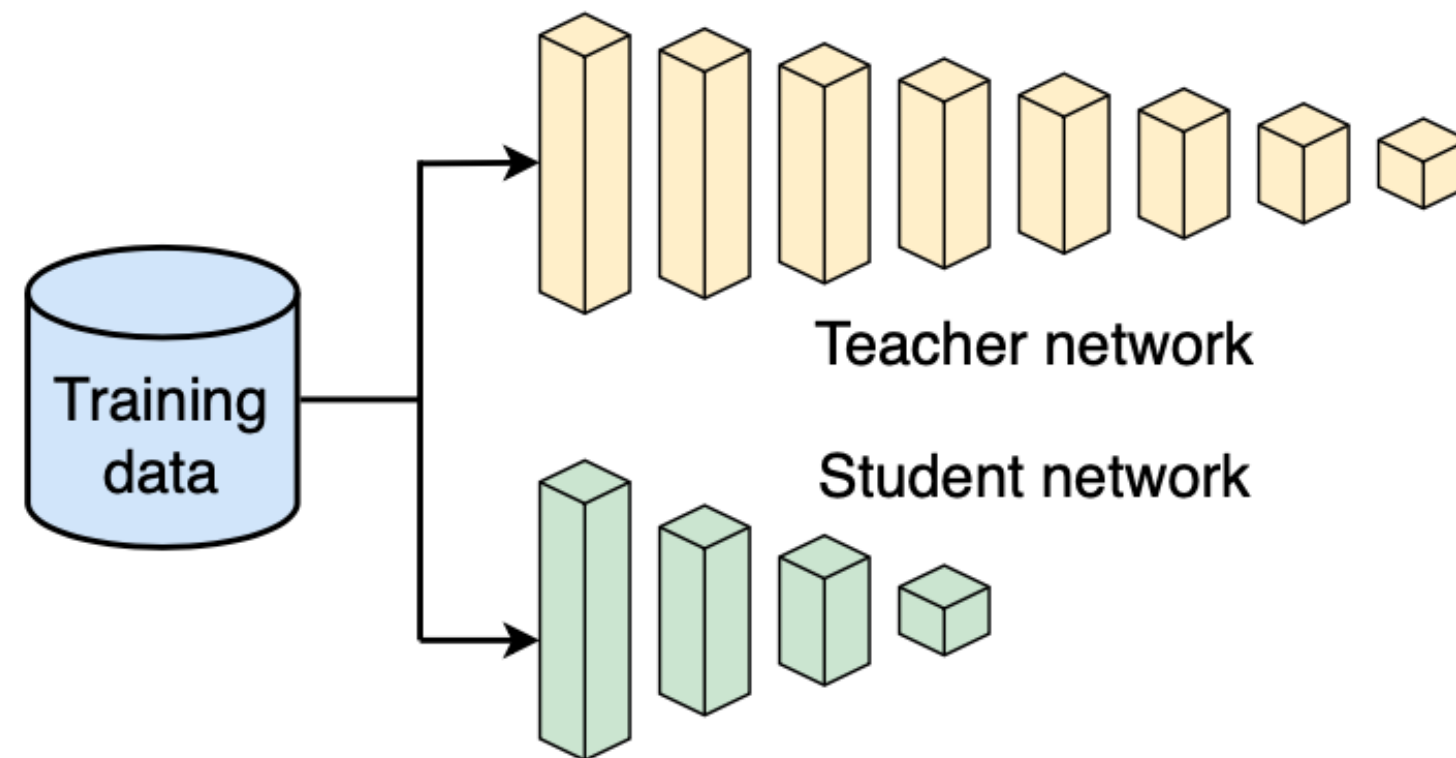
2.



## **Knowledge** *compression*

# Knowledge *compression*

El objetivo de *Knowledge Compression* es reducir el tamaño del modelo (*student network*) sin perder mucha precisión en comparación con el modelo original (*teacher network*). Esto es especialmente útil para implementar redes neuronales en dispositivos con recursos limitados, como teléfonos móviles o sistemas embebidos.



## Ventajas

- Reduce la memoria y los costos computacionales.
- Hace que los modelos sean más eficientes en edge devices.

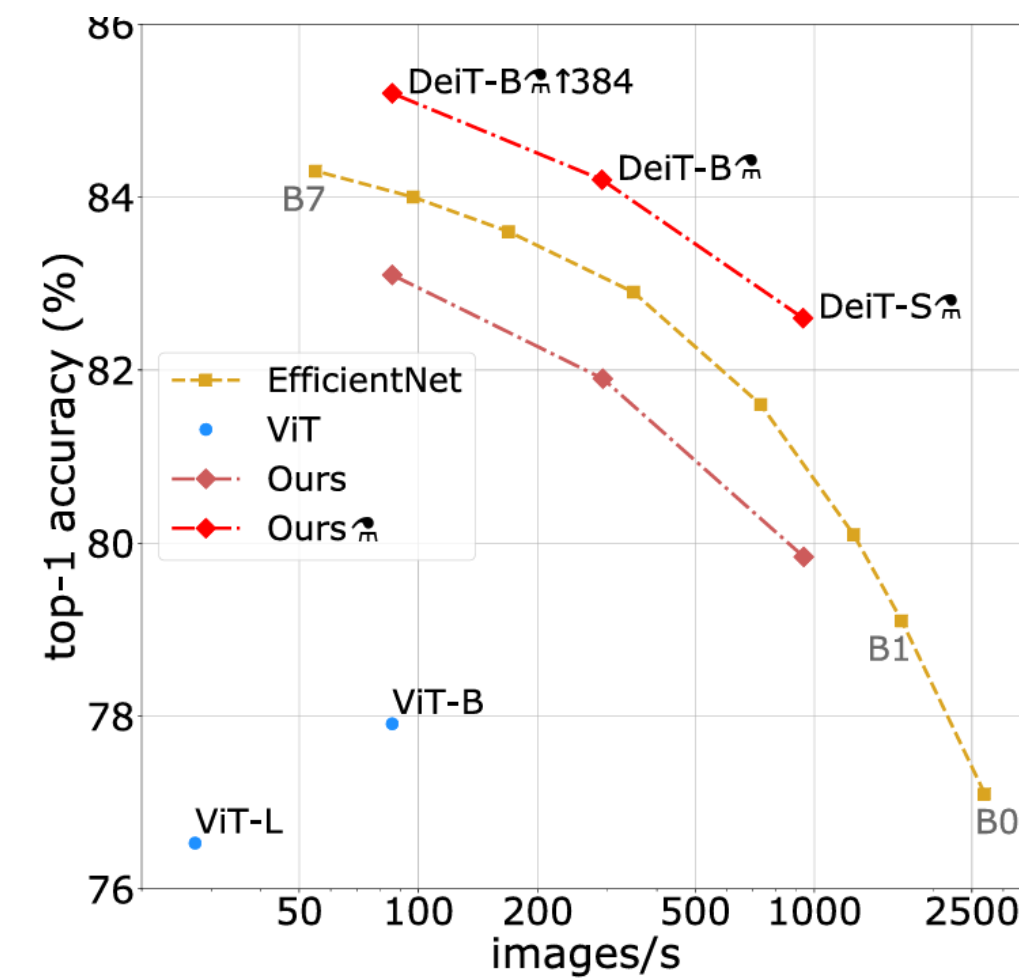
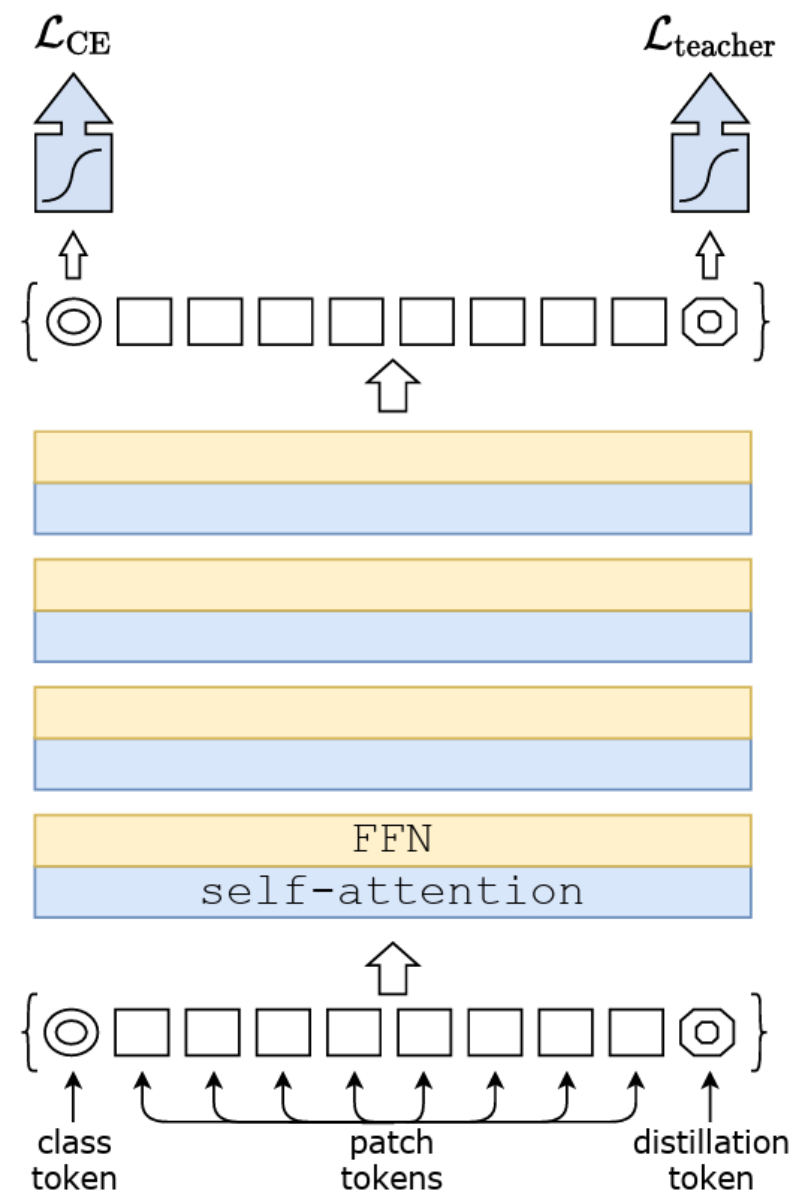
## Desafíos

- El *student* puede perder capacidad de generalización si la compresión es extrema.
- Puede ser difícil encontrar el balance entre rendimiento y eficiencia.



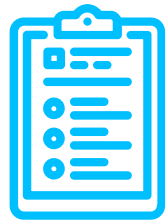
# DeiT

(Data-efficient image Transformers)





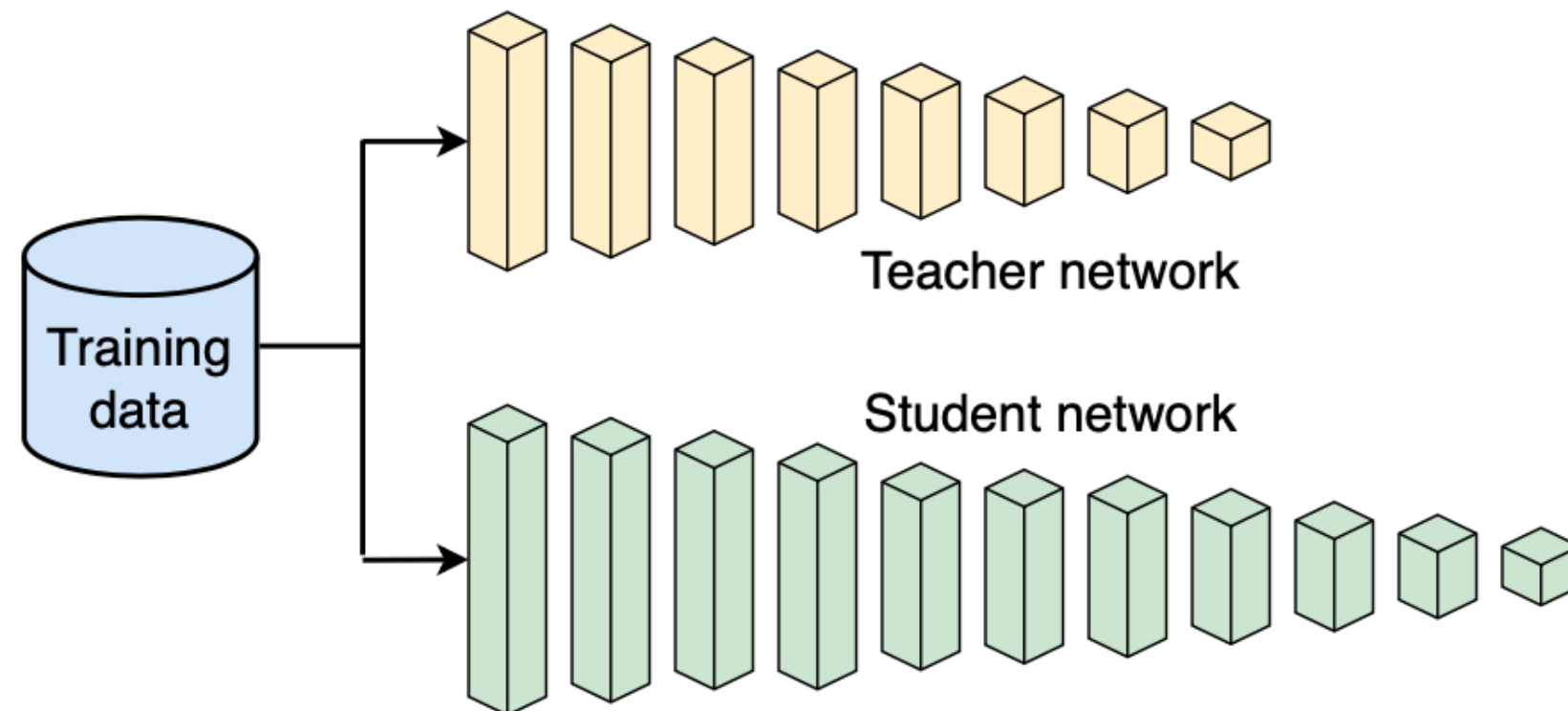
**3.**



**Knowledge** *expansion*

# Knowledge *expansion*

En contraste con *Knowledge Compression*, el objetivo de *Knowledge Expansion* no es reducir el modelo, sino **entrenar un *student network* que supere al *teacher network*** en términos de rendimiento y capacidad de generalización.



## Ventajas

- Permite que el *student* supere al *teacher* en tareas más complejas.
- Mejora la capacidad de generalización del modelo.

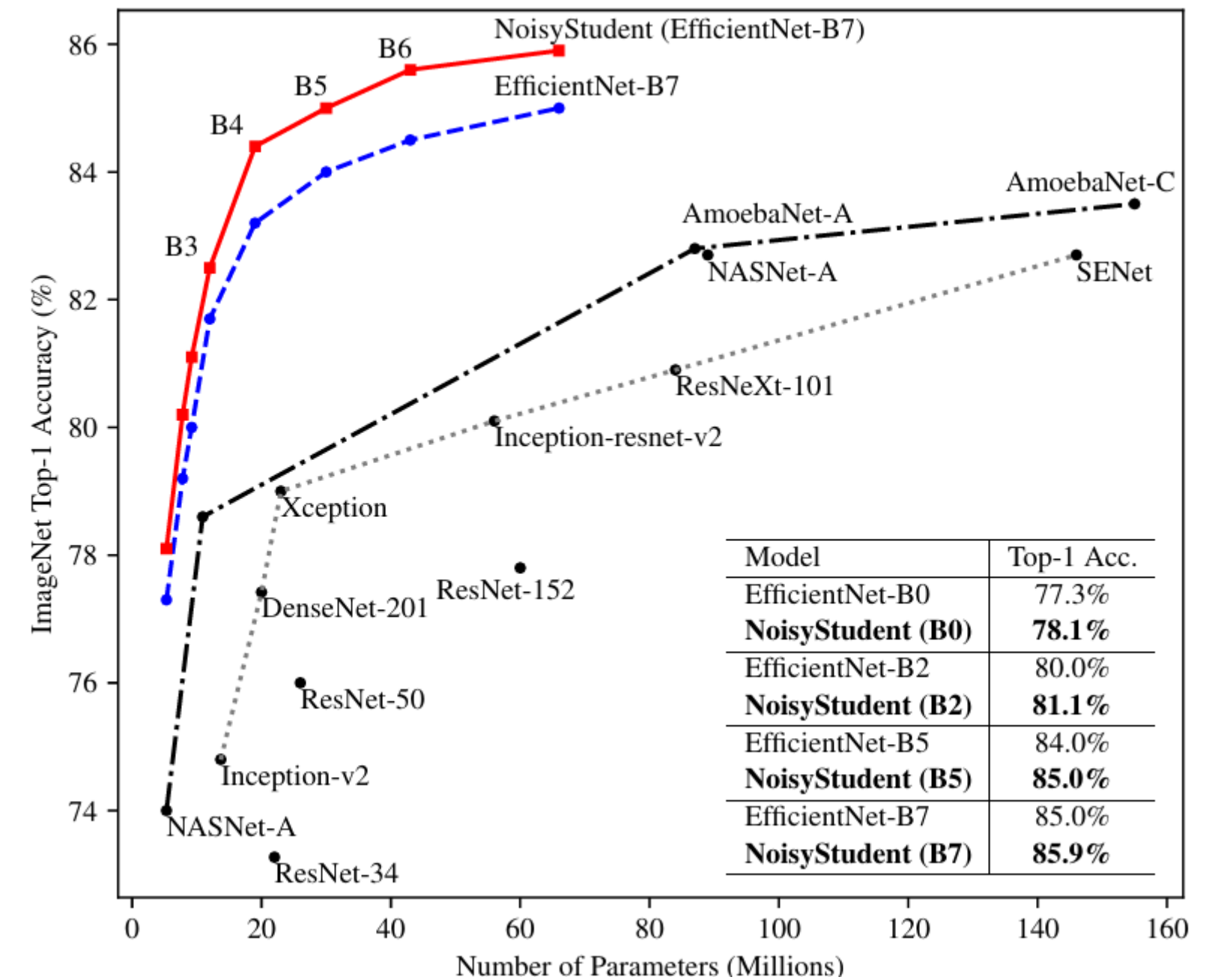
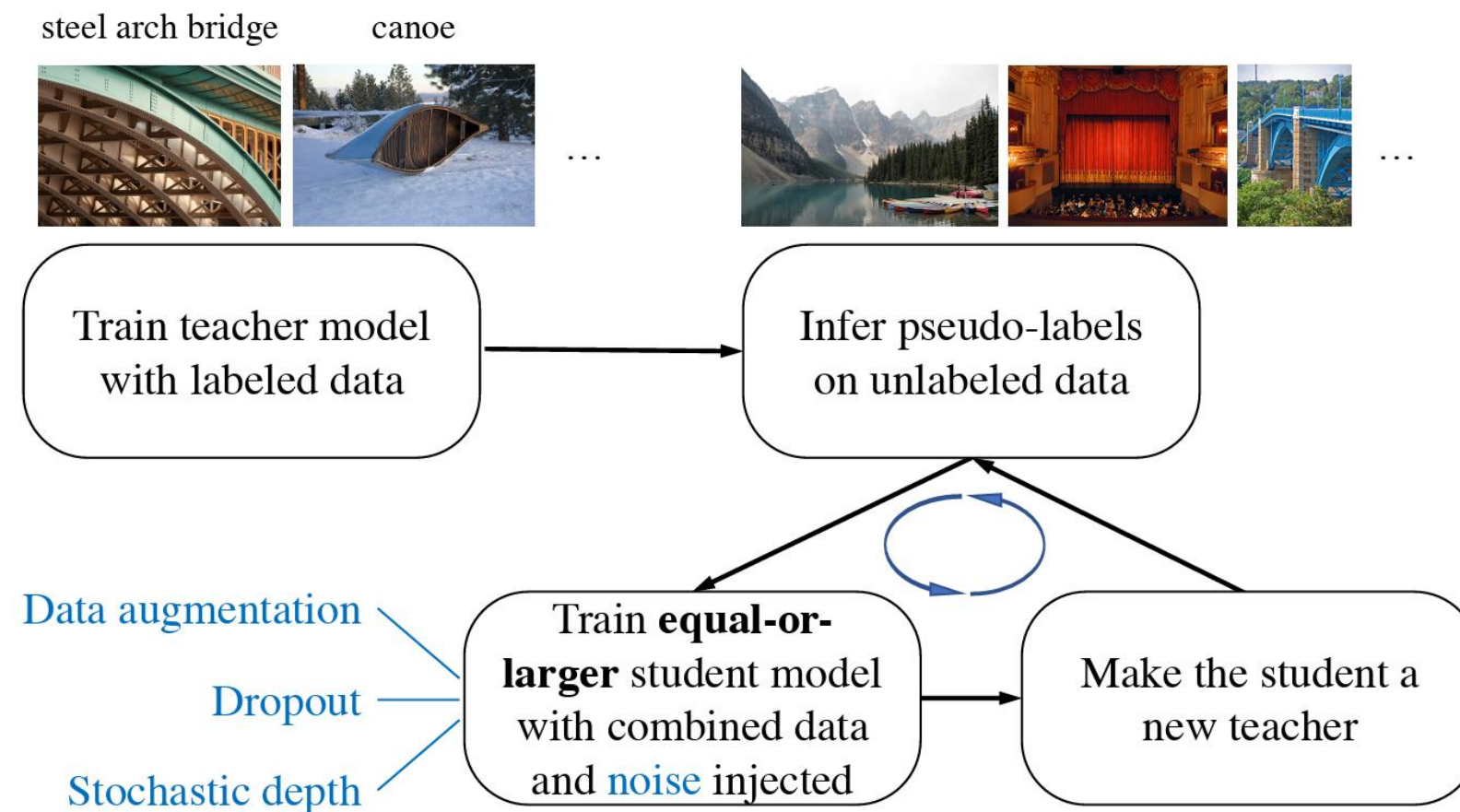
## Desafíos

- Si el *teacher* genera pseudo *labels* incorrectos, el *student* puede aprender información errónea.
- La expansión del conocimiento requiere más datos y cómputo.





# Noisy Student *Training*



# GRACIAS

*Victor Flores Benites*

