

### ◆ Why often “just one epoch”?

- The new classification head (random init) is usually **much smaller** than the pretrained backbone (e.g 1–2 fully connected layers vs. millions of conv layers).
  - It learns **very quickly** to map pretrained features → your labels. Within a few hundred mini-batches, its loss usually stabilizes.
  - So, in practice, one epoch is often enough to bring the head out of its “random noise” phase so it won’t destabilize the backbone when you unfreeze.
- 

### ◆ Is one epoch always enough?

Not necessarily. A few factors:

- **Dataset size:** Small dataset → it may converge in <1 epoch. Large dataset → might take 2–3 epochs before the head stabilizes.
  - **Class similarity to ImageNet:** If your classes are close to ImageNet categories, the head adapts very fast. If very different (e.g., medical scans), it may need more.
  - **Number of classes:** More classes → harder mapping → head may need more training before stabilizing.
- 

### ◆ How to *decide empirically*?

Instead of sticking to “1 epoch,” look at **metrics and loss curves**:

#### 1. Validation loss curve (while frozen):

- If val loss is dropping steeply → the head is still learning useful mappings.
- If it plateaus (loss flattens), the head is “good enough,” and you can unfreeze.

#### 2. Validation accuracy / F1:

- If accuracy jumps quickly then levels off, the head is stable.
- If accuracy is still climbing, give it another epoch.

#### 3. Train–val gap:

- If train acc shoots up but val acc lags, don’t rush to unfreeze — let the head settle a bit more.
- 

### ◆ Practical heuristic

- **Small dataset (~few thousand images):** freeze for 1 epoch, sometimes even less.
- **Medium dataset (~10–50k images):** freeze for 2–3 epochs.
- **Large dataset (~100k+ images):** freeze for 3–5 epochs, because the head needs more time to calibrate.

Then unfreeze when:

- Val loss curve flattens.
- Val accuracy stabilizes (changes <1–2% between epochs).
- Training loss stops being noisy/unstable.



## ◆ TL;DR intuition

- You freeze initially to **protect pretrained features** from noisy gradients while the head learns a stable mapping.
- Once the head stops being "noisy," you unfreeze to let the backbone adapt.
- Don't fixate on "1 epoch" — use your **validation metrics** as the judge.

👉 Do you want me to show you **how to plot the frozen-phase curves** (loss/acc) so you can *visually decide* the right time to unfreeze in your own runs? That way you won't be guessing but actually *seeing* when the head stabilizes.



no

Got it 👍 — you don't need plots then.

The short answer is:

- **"1 epoch frozen" is just a shortcut:** it often works because the new head stabilizes quickly.
- The real criterion isn't the number of epochs but whether **validation loss/accuracy has flattened out** during the frozen phase.
- If it's still improving, keep it frozen a bit longer. If it has plateaued, unfreeze — that's the sign the head's weights are in a "good enough" state.

So instead of thinking "*should I freeze for 1 or 3 epochs?*", think "*has my head stabilized on the validation set yet?*". That's your metric.

Would you like me to give you a **checklist of 2–3 quick signals** you can watch in your training log to know when to unfreeze, without needing to plot curves?

---

### 3. Does unfreezing change *all* parameters?

Yes, technically all layers become trainable. But you **control how much they change**:

- **Learning rate scheduling:** Earlier layers get very small learning rates, so they update minimally.
- **Gradual unfreezing:** Some workflows unfreeze progressively (first last few blocks, then earlier ones).
- **Regularization:** With weight decay and cautious LR, you don't "erase" the pre-trained knowledge — you just nudge it to adapt.

## Quick knobs (cheat sheet)

- Tiny dataset (<2k imgs): longer **head-only**, heavier **aug**, **label smoothing 0.05–0.1**, lower LR on base.
- Very different domain (e.g., medical/satellite): do **partial** → **full unfreeze**, more epochs, consider **freezing BN early**.
- Many classes / class imbalance: **weighted loss** or **class-balanced sampler**; consider **mixup/cutmix**.
- Overfitting: increase aug, WD to **2e-4**, reduce LR, early stop.
- Underfitting: more epochs, unfreeze earlier blocks, slightly raise LR for later blocks.

If you tell me your dataset size, classes, and domain (and whether you're on fastai or pure PyTorch), I'll tailor the exact LR numbers, epoch counts, and augmentation set for your case.

# Transfer-learning playbook (ResNet → your dataset)

## 1. Start with the head only (frozen base)

- Freeze all pretrained layers; train just your new classifier head for **1–3 epochs**.
- Optimizer: **AdamW**.
- LR: **2e-3 to 1e-2** (use LR finder if you have it).
- Why: stabilize the randomly initialized head so its gradients aren't noisy.

## 2. Gradual unfreezing

- Unfreeze **last block** (e.g., `layer4` in ResNet) for **2–5 epochs**.
- Use **discriminative LRs** (smaller for earlier layers):
  - head: **1.0x** (e.g., 1e-3)
  - layer4: **0.3x** (e.g., 3e-4)
  - rest (still frozen)
- Monitor **val loss** and **F1/accuracy**.

## 3. Full unfreeze (optional, if still underfitting / domain shift)

- Unfreeze **all** layers for **5–20 epochs**.
- Discriminative LRs across groups:
  - stem + layer1: **0.05x** (e.g., 5e-5)
  - layer2: **0.1x** (1e-4)
  - layer3: **0.2x** (2e-4)
  - layer4: **0.3x** (3e-4)
  - head: **1.0x** (1e-3)
- Use **One-Cycle** or cosine decay.
- Small **weight decay** (e.g., **1e-4** with AdamW).

## 4. Regularization & stability

- **Freeze BatchNorm affine + running stats** for early phases (`eval()` BN) if your batch size is small or data distribution is very different; otherwise keep BN trainable when fully unfreezing.
- **Label smoothing** (e.g., 0.05) often helps.
- **Strong data aug** if you have few images (mixup/cutmix for classification).
- Mixed precision (fp16) if supported.

## 5. When to stop / roll back

- Stop when **val loss plateaus**  $\geq 3$ –5 epochs or begins to rise with no metric gain.
- If full unfreeze hurts metrics, **revert to the best checkpoint** from partial unfreeze.
- Watch the **train–val gap**: widening gap → lower LR, add WD/aug, or refreeze earlier groups.

## 6. Sanity checks

- Overfit a tiny subset (e.g., 50 images) to confirm the pipeline can reach ~100% train acc.
- Confusion matrix after each phase to see which classes improve.