

ACM
Multimedia 2026
Rio de Janeiro, Brazil

Post-hoc Image Watermarking

From training recipes to a hands-on Colab

Adobe

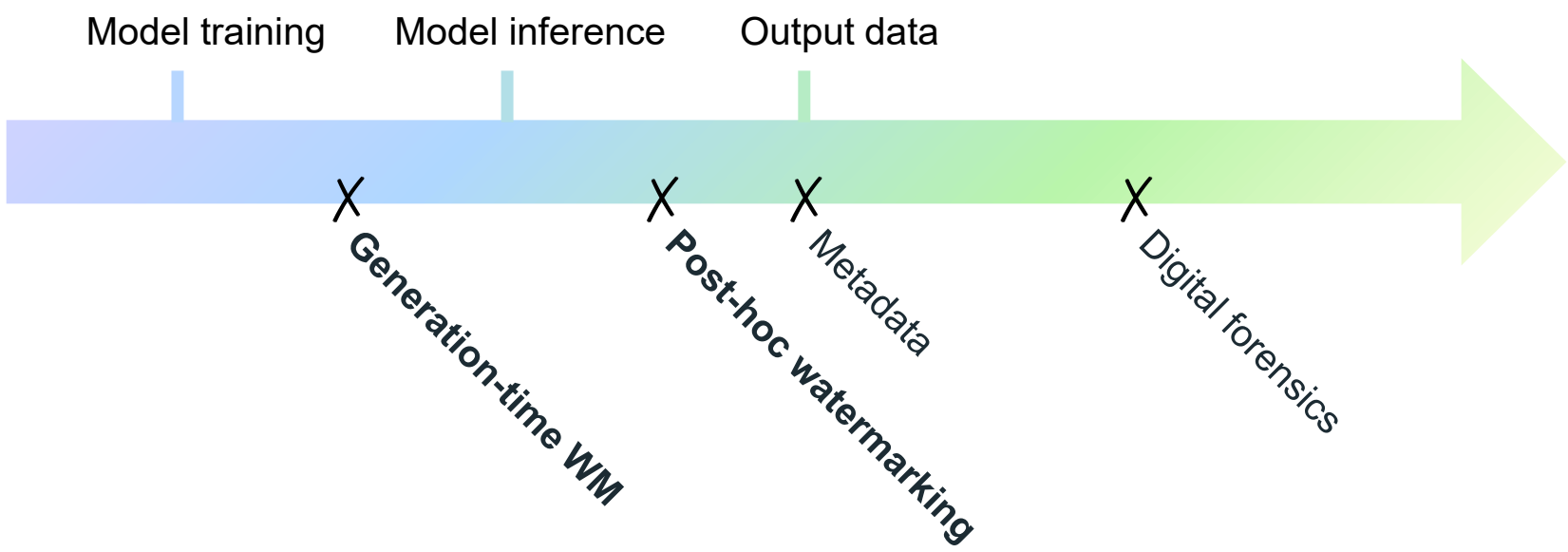
Meta

UNIVERSITÀ DI SIENA 1240

UNIVERSITY OF MARYLAND

Watermarking can be done at different stages

- How is post-hoc watermarking trained and used in practice?
- What are the key design choices? Let’s build one!



Adobe Meta Università di Siena 1240 University of Maryland

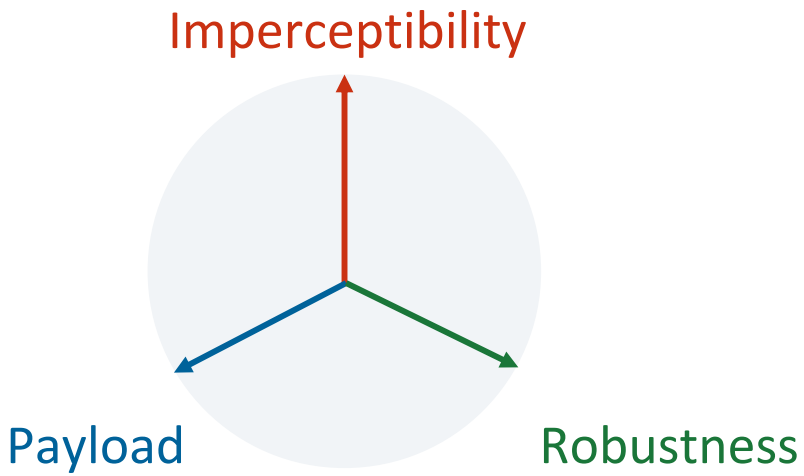
Outline

- Post-hoc image watermarking: how it works
- Practical recipes: resolution, imperceptibility, robustness, stable training, localization
- Hands-on: train your own image watermark in a Colab (NanoMark)

1. Post-hoc Image Watermarking

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

Reminder: 3 criteria



1. Imperceptibility

   Distortion must be low

2. Payload

   The message to hide can be long enough

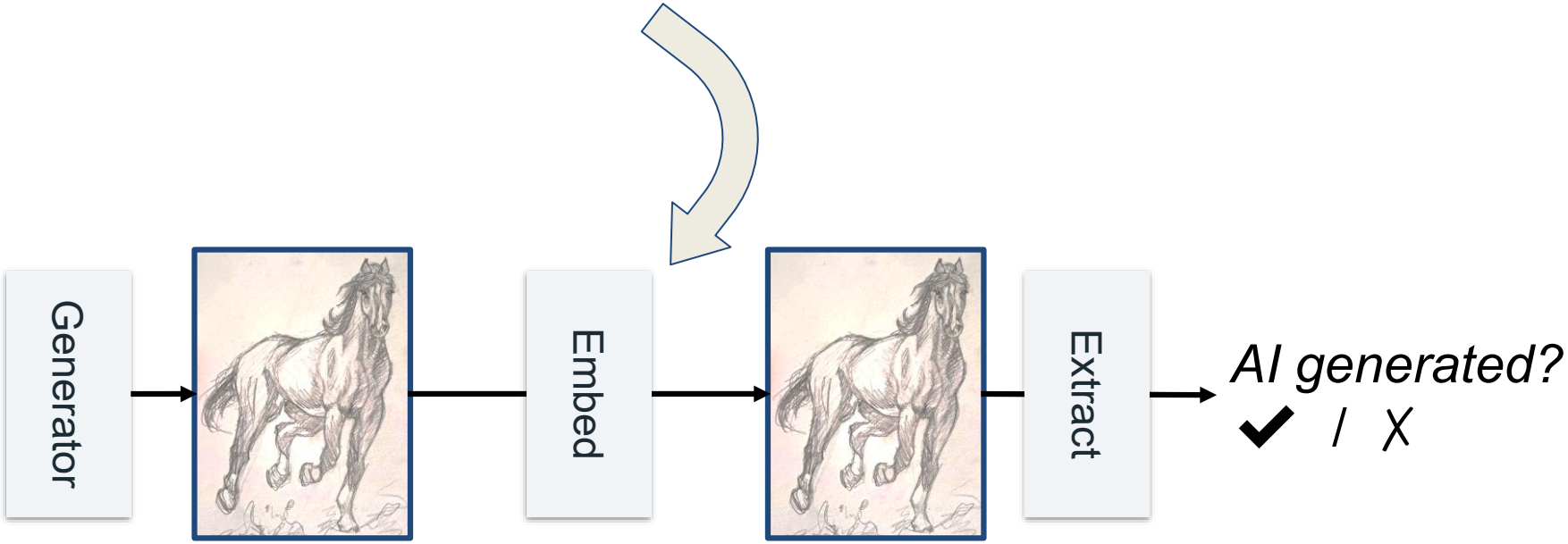
   ↔ can be detected with high confidence

3. Robustness

   The message must be recovered even when the content is edited

Post-hoc watermarking

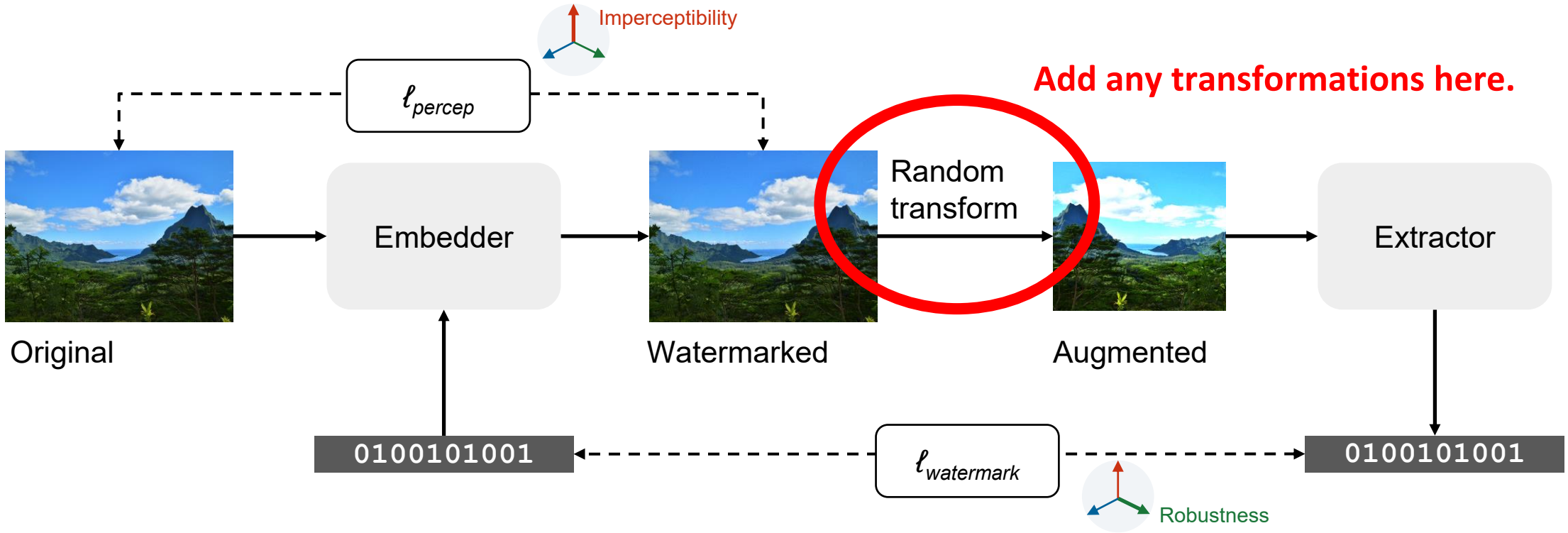
- Watermark embedder = whatever you want



Adobe Meta Università di Siena 1240 University of Maryland

Watermarking with deep neural networks

- 2 categories of losses.



Adobe Meta Università di Siena 1240 University of Maryland

How do we do detection?

Generated by our model



Hidden message
m' : **11111…11**

Ext.

m : **11111…01**

→ **96 / 100**

Natural image

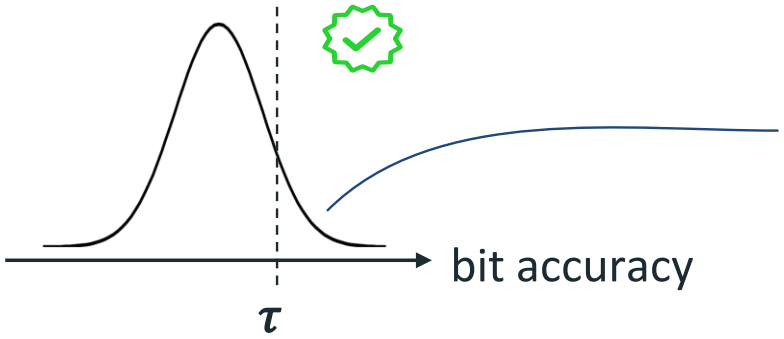


Ext.

Random *m*: **10100…11**

→ **54 / 100**

Test: bit accuracy (m,m') > τ ex: τ = 80%

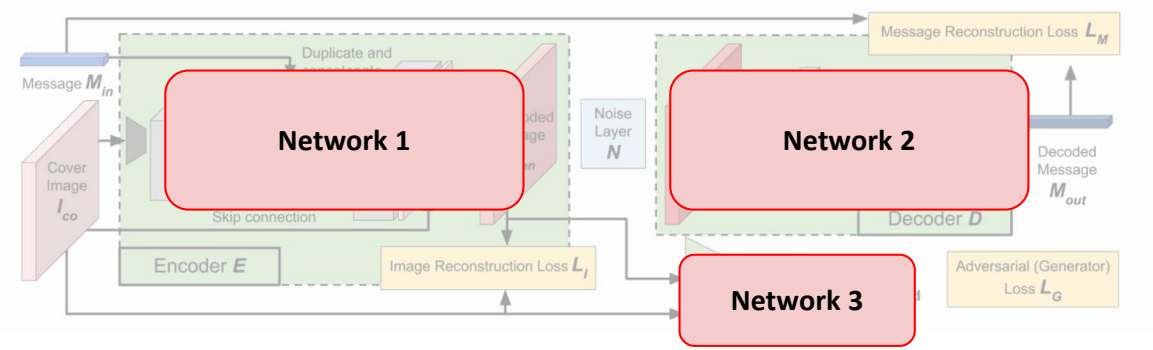


$$\mathbb{P}\left(M(m,m') \geq \tau \mid H_0\right) = \frac{1}{2^{n_{\text{bits}}}} \sum_{i=\tau+1}^{n_{\text{bits}}} \binom{n_{\text{bits}}}{i}$$

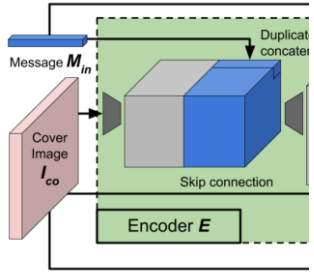
Adobe Meta Università di Siena 1240 University of Maryland

Encoder/decoder: some issues

- Training is very unstable, hard to reproduce.



- Image quality is ok, but not enough to be used in products.
- Cannot scale to a lot of bits because of how the message is processed.



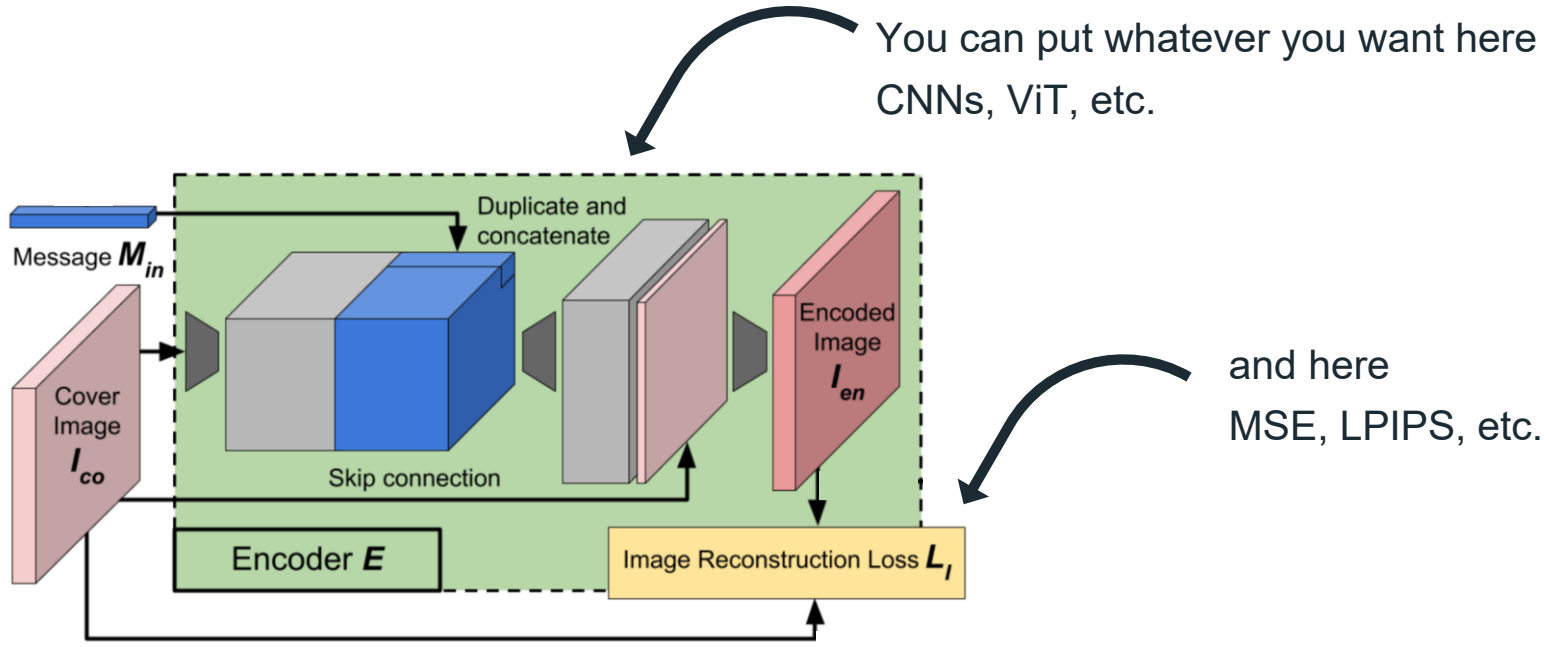
- Cannot scale to high resolution (same reason).

- Etc.

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

First example: HiDDeN

- [📄 Zhu et al., HiDDeN: Hiding Data with Deep Networks. In ECCV, 2018.]



Newer works

Pushing state-of-the-art in encoder-decoder based approaches

- TrustMark
- SynthID-Image
- PixelSeal
- InvisMark

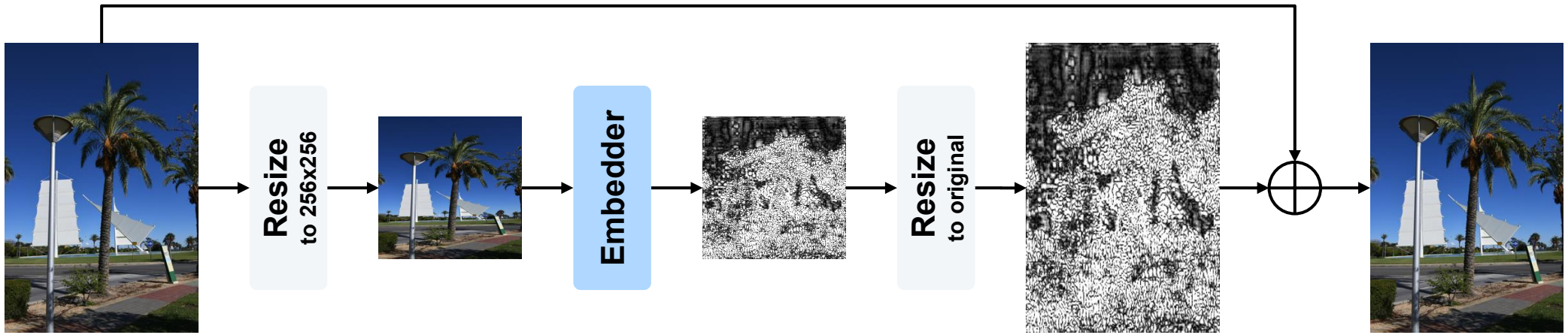
New capabilities: localized methods, forgery detection, scaling bits, etc.

- Watermark-Anything
- MaskWM
- Swift
- ChunkySeal

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

How to operate at any resolution?

- Make the embedder/extractor work at fixed resolution (e.g. 256x256):

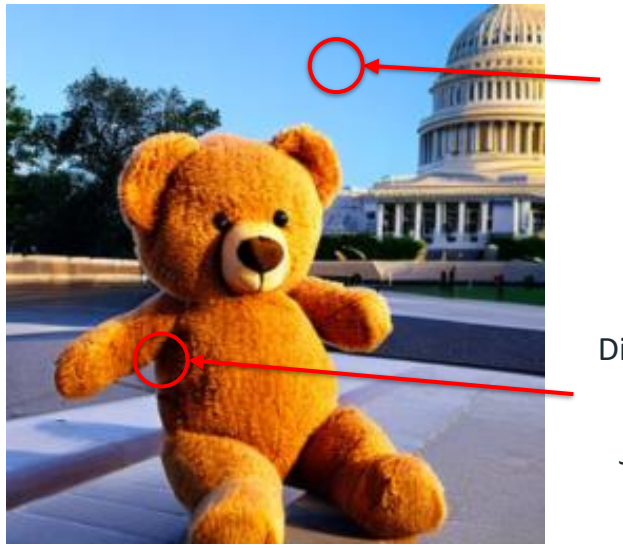


- Train at fixed resolution. Works for almost all of the models out-of-box *at inference time*.
- You can finetune/train with images at different sizes to improve results

[TrustMark]

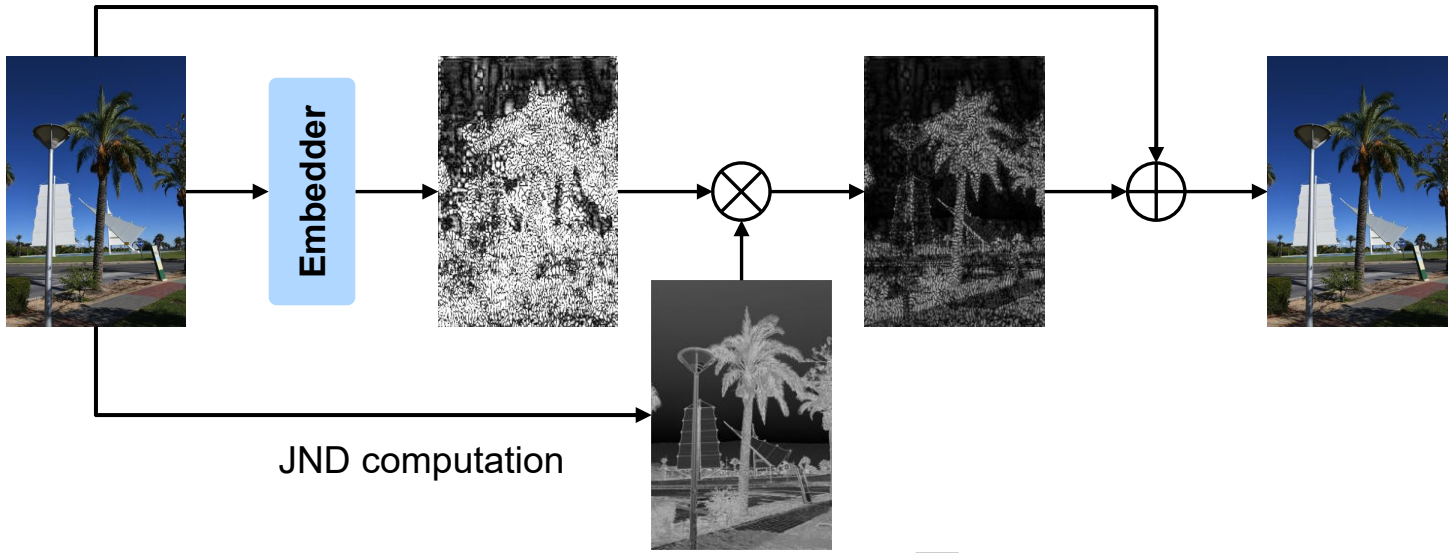
How to make it imperceptible?

- No perceptual losses really captures human perception (MSE, LPIPS, FocalLoss, etc.).
- Some recent works use Just Noticeable Difference (JND) heatmaps



Difference would be visible
JND should be ≈0

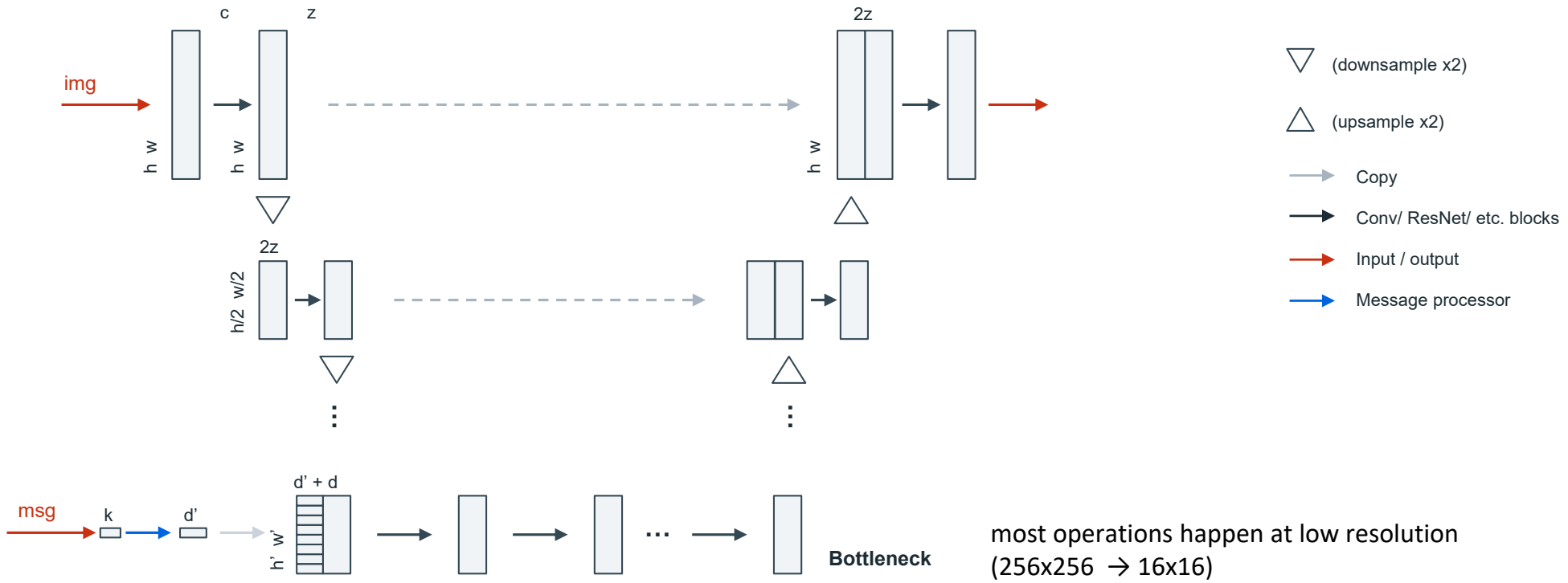
Difference would be hard to perceive
JND should be >0



[Watermark Anything]

What are the best architectures?

- **Embedder**: UNet / architectures from latent AE



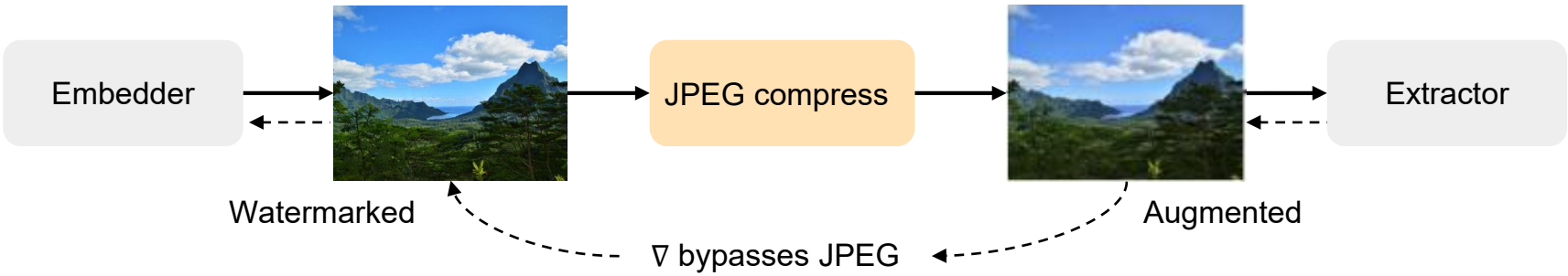
most operations happen at low resolution
(256x256 → 16x16)

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

How to make it robust to compression?

- Use straight-through estimator for JPEG - pytorch pseudocode:

```
jpeg_out = JPEG(watermarked)
jpeg_ste = watermarked + (jpeg_out - watermarked).detach()
```



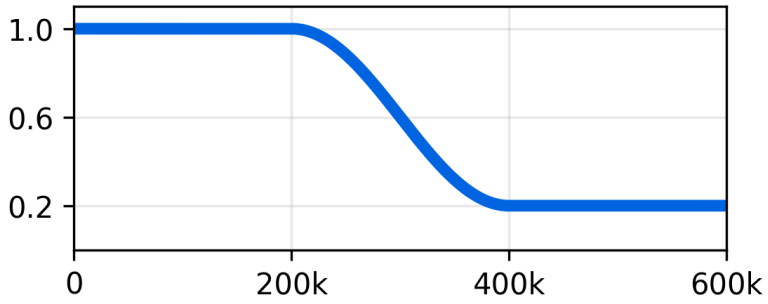
- Works surprisingly well, and across compression methods and modalities

[Zhang et al., Towards Robust Deep Hiding Under Non-Differentiable Distortions for Practical Blind Watermarking. In MM, 2021.]

How to make training “stable”?

- **Training is unstable**. In many cases:
  - training does not start at all and stagnates at 0.5 bit accuracy,
  - it changes a lot depending on random seeds,
  - it converges to different minima.

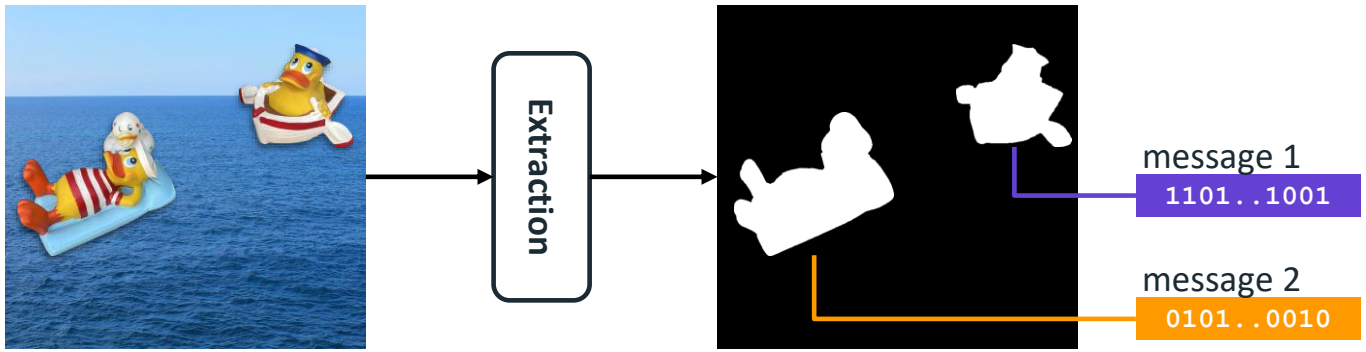
- **Curriculum learning** (start with something more easy, gets harder during training). E.g.:
  - Start without augmentations, add them at half of the training
    [📄 Distortion agnostic deep watermarking]
  - Start with a multiplicative factor on the watermark distortion, that decreases with time.
    [📄 PixelSeal]



Adobe Meta UNIVERSITÀ DI SIENA 1240

How to make it localized?

- Make the extractor predict a detection mask + messages



Edited
(change background, inpaint, crop, etc.)

Predicted detection mask

- e.g. to know which part has been edited, which part is watermarked

[Watermark Anything]

[MaskWM]

2. Hands-on: NanoMark Colab

NanoMark

Link to the colab:
https://colab.research.google.com/drive/18MH6fttuPWcGA3BriDTV0diO6B9Dkoek?usp=sharing



Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

Hands-on: what is in the notebook

- Run it: open the link, set the runtime to a T4 GPU (free tier), Run all
- Setup & data: Imagenette, train at 128x128, keep a native-resolution validation set
- JND heatmap: where the watermark can hide (edges, textures)
- Models: tiny UNet embedder (bits injected at the bottleneck) + CNN extractor
- Strength curriculum: start with a strong residual, anneal to a high-PSNR target
- Training (600 steps): bit accuracy goes up while PSNR stays high
- Robustness at native resolution: JPEG, noise, blur, resize, crop, flip, brightness
- Inference: predict a 128x128 residual, upsample, apply JND at full resolution

Entry Points

github.com/adobe/trustmark
(inference)

github.com/facebookresearch/videoseal
(inference)(training)

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND

Conclusion & Key Takeaways

- Post-hoc watermarking is the de facto standard in industry
  - Easy to deploy, very modular;
  - Easy to adapt to other modalities (video, audio);
  - Still a lot of room for improvement (architectures, optimal training schemes, etc.).
- What makes it work in practice:
  - JND masks, fixed-resolution embedding + upsampling, compression-aware augmentations;
  - a curriculum on watermark strength (all in the NanoMark Colab).

Adobe Meta UNIVERSITÀ DI SIENA 1240 UNIVERSITY OF MARYLAND