

VQ-TRANSPLANT: Efficient VQ-Module Integration for Pre-trained Visual Tokenizers

Xianghong Fang¹ · Yuan Yuan² · Dehan Kong¹ · Tim G. J. Rudner^{1 3}

¹University of Toronto · ²Boston College · ³Vijil

Plug-and-play VQ replacement

→ No retraining of encoder/decoder

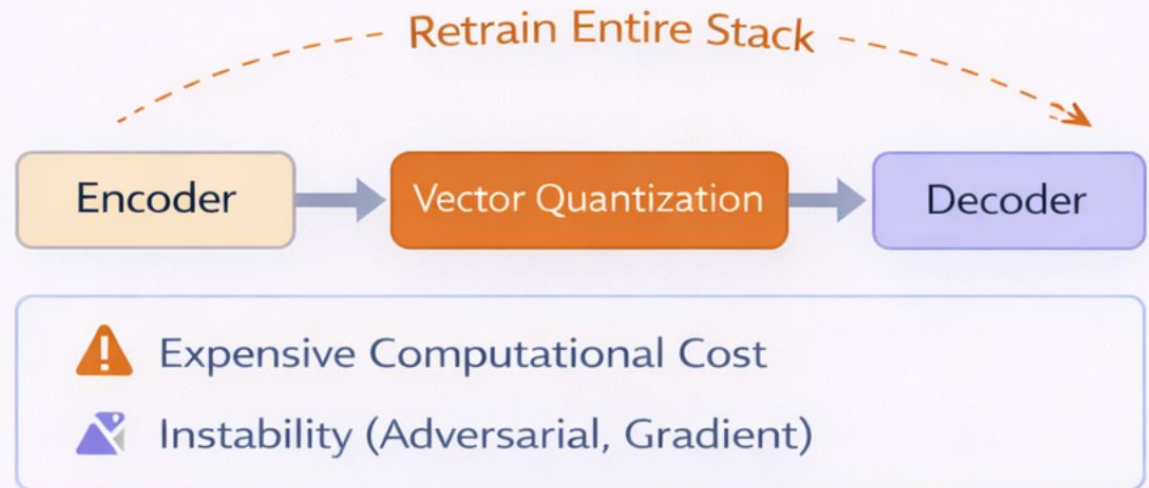
→ Up to **21.8x less compute**

Maintains or improves reconstruction quality (r-FID)

<https://github.com/VQ-Research/VQ-Transplant>

Problem

- Improving **VQ** requires **retraining** the entire tokenizer
- Expensive and **unstable** (adversarial / gradient issues)



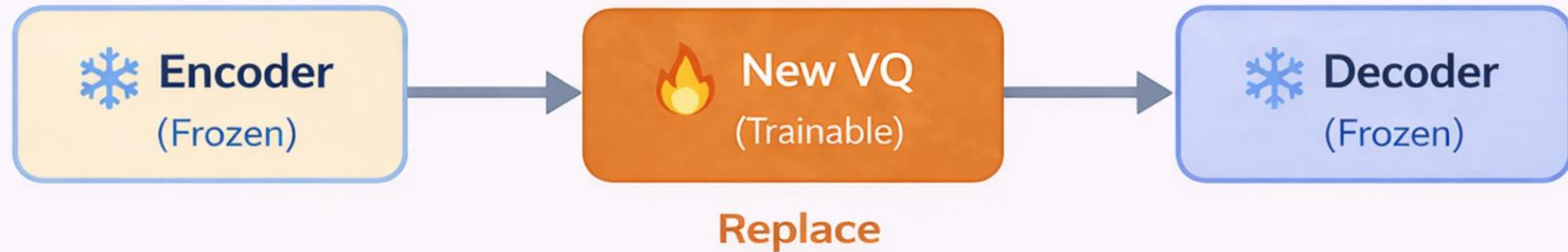
Can we improve the VQ module
without retraining the entire system system?



Core Idea

Swap VQ. Freeze Everything Else.

- Freeze encoder + decoder*
- Only **replace** VQ



Keep the pretrained encoder and decoder fixed, and **only train a new VQ module.**

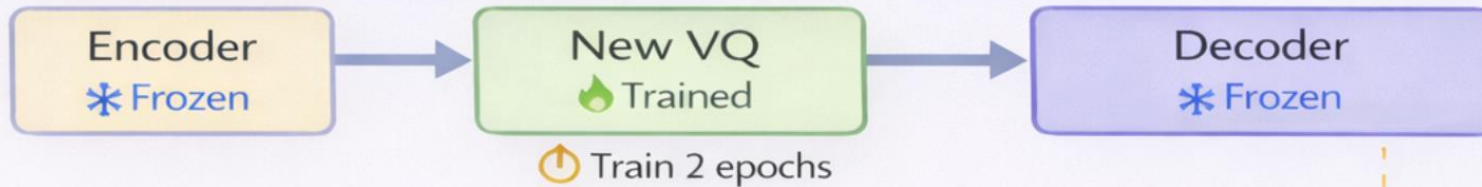
A modular, plug-and-play design
to **efficiently improve VQ.**

Method

Our method consists of two simple steps.

1 Step 1: VQ Module Substitution

- Replace the original VQ with a new one, while freezing encoder and decoder



2 Step 2: Lightweight Decoder Adaptation

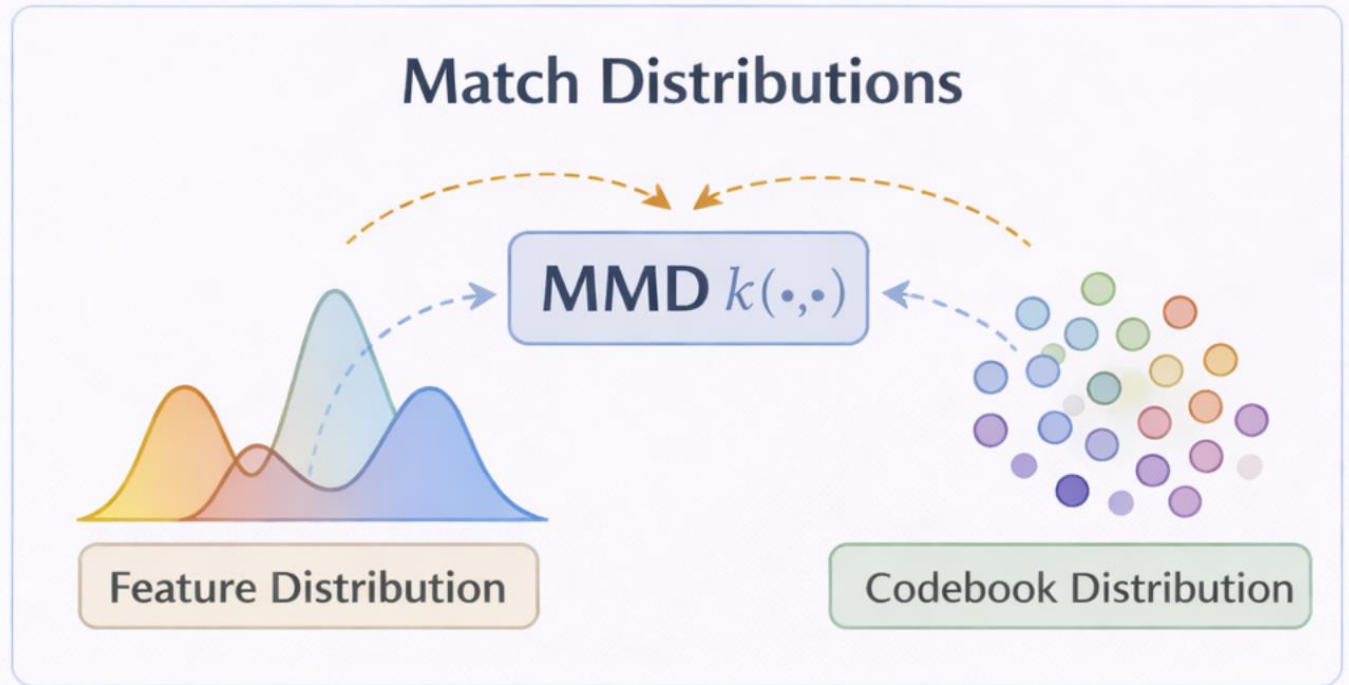
- Fine-tune the decoder for a few epochs to restore alignment **Decoder adaptation**



We update **components sequentially**, not the **entire system**.

MMD-VQ

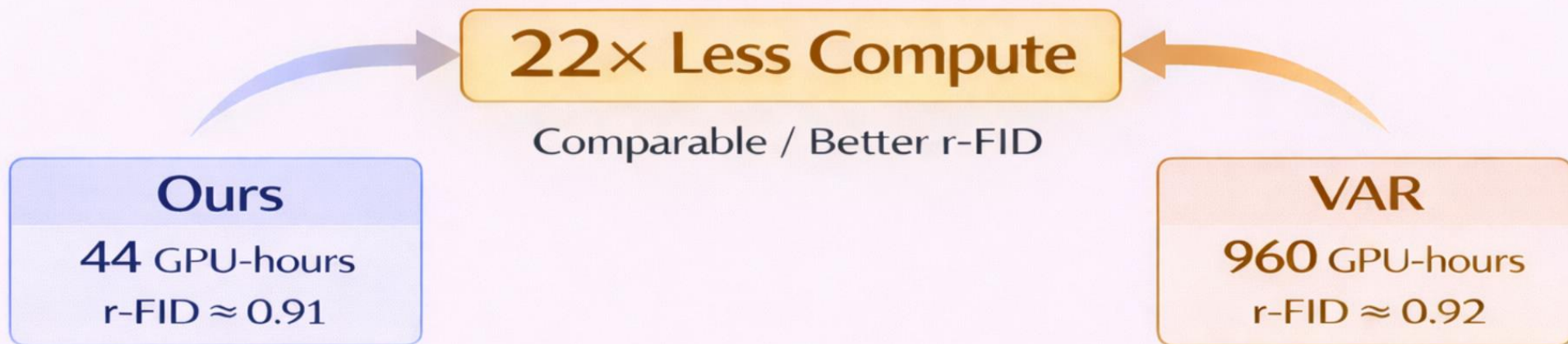
- Distribution-aligned quantization (MMD-VQ)



Align **codebook** distribution with **feature** distribution.

Results

- Comparable or better r-FID with significantly less compute



Models	GPU Hours	r-FID
VAR	960	0.92
VQ-Transplant-a	44	0.91
VQ-Transplant-b	79	0.87
VQ-Transplant-c	114	0.82
VQ-Transplant-d	149	0.79

Insight & Conclusion



Lower quantization error alone is **not sufficient** after VQ replacement.

Decoder-latent mismatch degrades reconstruction quality.



Lightweight decoder adaptation effectively resolves the mismatch

- ✓ Plug-and-play VQ replacement
- ✓ Avoids end-to-end retraining



“VQ-Transplant enables **compute-efficient** improvement of visual tokenizers.

Thank You!

Visit our project VQ-Transplant



github.com/VQ-Research/VQ-Transplant