DiVeQ: Differentiable Vector Quantization Using the Reparameterization Trick

Mohammad Vali Tom Bäckström Arno Solin

ellis INSTITUTE Aalto University

GitHub PyPI

TL;DR

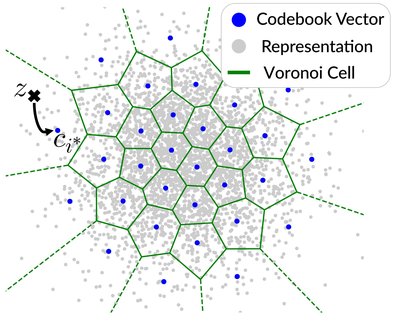
DiVeQ is a novel technique that allows end-to-end training of Vector Quantization (VQ) in DNNs, while avoiding common pitfalls of state-of-the-art methods and achieving superior performance. DiVeQ acts as a *drop-in replacement* for standard VQ layers without requiring any auxiliary losses or hyperparameter tuning.

Background

- **Vector Quantization (VQ)** models an abstract discrete representation of a distribution via codebook vectors.

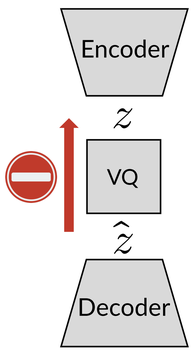
$\hat{z} = \arg\min_{c_i} \|z - c_i\|_2 = c_{i^*}$

c_{i^*} is the closest codebook vector.



- **Gradient Collapse Issue:** VQ is not differentiable.

$\frac{\partial \hat{z}}{\partial z}$ does not exist ⇒ backpropagation fails!



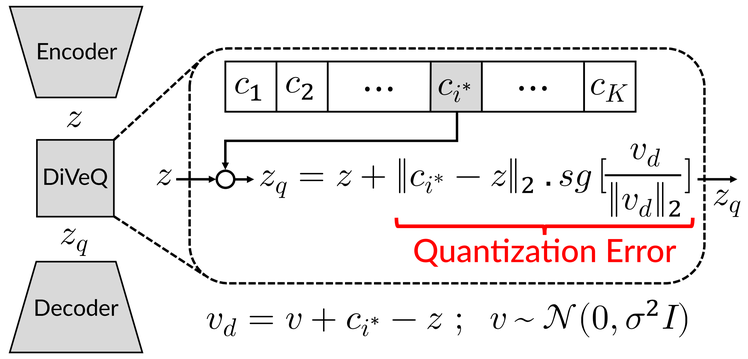
- **Previous Solutions (state-of-the-art)**
  1. Straight-Through Estimator (STE)
  2. Exponential Moving Averages (EMA)
  3. Rotation Trick (RT)
  4. Straight-Through Gumbel-Softmax (ST-GS)
  5. Noise Substitution Vector Quantization (NSVQ)

- **Research Gap:** Shortcomings of previous solutions.

	STE	EMA	RT	ST-GS	NSVQ	DiVeQ	SF-DiVeQ
No auxiliary loss terms	✗	✗	✗	✗	✓	✓	✓
No hyperparameter tuning	✗	✗	✗	✗	✓	✓	✓
Unbiased codebook gradients	✗	N/A	✓	✗	✓	✓	✓
No train–test mismatch	✓	✓	✓	✗	✗	✓	✓
End-to-end training	✗	✗	✗	✓	✓	✓	✓
Precise quantization mapping	✓	✓	✓	✓	✗	✓	✓
No codebook misalignment	✗	✗	✗	✗	✗	✗	✓
Avoiding codebook collapse	✗	✗	✗	✗	✗	✗	✓

DiVeQ

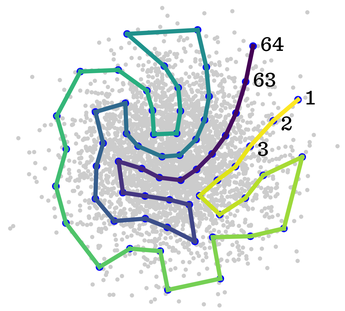
- **DiVeQ** models VQ as addition of a simulated quantization error.



- **Valid Gradients** ⇒ $\frac{\partial z_q}{\partial z}$: Valid ✓ and $\frac{\partial z_q}{\partial c_{i^*}}$: Valid ✓

Space-Filling DiVeQ

- **Space-Filling Vector Quantization** is a modification of VQ that quantizes the latent to a piecewise continuous curve constructed by the line-segment connections between adjacent codebook vectors.



- **Space-Filling DiVeQ** is similar to DiVeQ, but maps z to a random point on the line connecting two adjacent codebook vectors.

Quantization Error

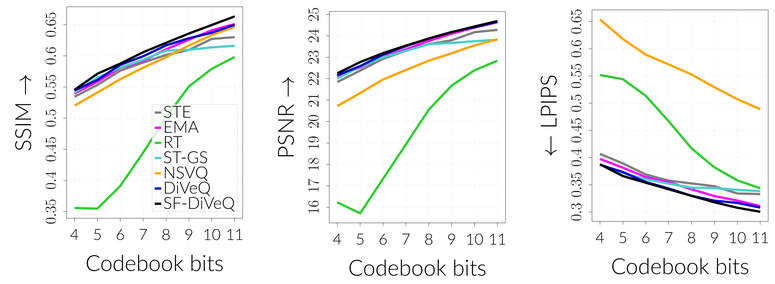
$$z_q = z + \|c_{i^*} - z\|_2 \,.\, sg\left[\frac{(1-\lambda_{i^*})v_{d_{i^*}}}{\|v_{d_{i^*}}\|_2}\right] + \|c_{i^*+1} - z\|_2 \,.\, sg\left[\frac{\lambda_{i^*}\ v_{d_{i^*+1}}}{\|v_{d_{i^*+1}}\|_2}\right]$$

$$\begin{cases} v_{d_{i^*}} = c_{i^*} - z + v \\ v_{d_{i^*+1}} = c_{i^*+1} - z + v \\ v \sim \mathcal{N}(0, \sigma^2 I) \end{cases}$$

c_{i^*} — $\lambda_{i^*} \in U(0,1)$ — c_{i^*+1}

Experiments

- **Experiment 1:** Image compression via VQ-VAE



- DiVeQ and SF-DiVeQ consistently improve image reconstruction quality across different codebook sizes.

- **Experiment 2:** Image generation via VQGAN

FID↓	Codebook bits			
	8	9	10	12
STE	6.64	5.57	**5.28**	6.69
EMA	6.86	6.30	6.32	6.24
RT	9.32	7.55	6.40	**5.44**
ST-GS	8.47	6.81	5.48	5.47
NSVQ	81.5	70.4	59.2	48.9
DiVeQ (ours)	**5.90**	6.69	6.32	7.69
SF-DiVeQ (ours)	6.24	**5.21**	5.57	6.00

- DiVeQ and SF-DiVeQ improve generation quality, especially for challenging small codebook sizes.

Easy to Use

```
pip install diveq

from diveq import DIVEQ
vector_quantizer = DIVEQ(num_embeddings, embed_dim)
```


mohammad.vali@aalto.fi