Continuous Deep Embed for Vision Transformers

Codebook-Gated MLPs as a Continuous Analog of Deep Embedding

Mitchell Mosure
Independent Researcher
mitchell@mosure.me

August 2025

Abstract

RWKV-8 "Heron" reports a discrete *deep embed* mechanism: token-dependent embeddings injected across depth to raise capacity without widening the backbone. This paper develops a vision counterpart, **Continuous Deep Embed (CDE)**, for Vision Transformers (ViTs). CDE attaches a token-conditioned, codebook-based gate to MLP branches and scales activations multiplicatively. Assignments use cosine similarity over a codebook, implemented as either a soft top-k mixture or a hard (vector-quantized) selection with a straight-through estimator. The design yields predictable parameter and compute overheads, preserves the ViT block topology, and integrates with standard training schedules.

We give closed-form overhead formulas, stabilization schedules (identity start, temperature annealing, small top-k), and a reproducible harness with per-configuration checkpoints, MLflow logs, and ImageNet preparation scripts. On ImageNet-scale settings, CDE matches or modestly improves ViT-S/16 accuracy at low cost. Results are reported for both width-gated and expanded-hidden variants. Source: github.com/mosure/continuous_deep_embed.

1 Introduction

Depth-wise conditioning raises representational capacity by allowing per-token signals to modulate intermediate computations. Reports for RWKV-8 "Heron" describe a discrete *deep embed* table that injects token-specific vectors through the residual stream, improving quality–efficiency trade-offs in sequence models. This motivates a ViT-oriented design that preserves block structure while enabling token-aware modulation.

This paper proposes *Continuous Deep Embed (CDE)*: a codebook-gated scaling of MLP branches in ViT blocks. CDE computes token–code similarities, mixes a small number of code vectors, and scales activations multiplicatively. Two assignment regimes are considered: soft top-k mixtures and hard vector quantization (VQ) with a straight-through estimator (STE). The approach adds a transparent overhead and is orthogonal to attention modifications. We follow standard reporting practices (e.g., Vaswani et al. [10]) to facilitate comparison.

Contributions.

1. **Continuous deep embedding for ViTs.** A token-conditioned gate that scales MLP activations using a codebook; compatible with DeiT/ViT blocks.

2. **Closed-form accounting.** Parameter and FLOP expressions for assignment, mixing, and application under assign-once and per-layer regimes.

3. **Stabilization schedules.** Identity start and temperature annealing that stabilize assignments and improve early convergence.

4. **Reproducibility.** An arXiv-friendly codebase with per-configuration checkpoints (best/last), MLflow logging, and license-respecting ImageNet preparation.[1]

2 Related Work

Vision Transformers. ViTs [1] and DeiT [8] established patch-based attention for image recognition.

Conditional computation and experts. Sparse MoE [2, 7] scales capacity via routed experts. CDE keeps one MLP per block and modulates it using a compact codebook mixture, avoiding routing and load balancing.

Vector quantization and relaxations. VQ-VAE [9] learns discrete codebooks; Gumbel-Softmax [3, 4] provides differentiable relaxations. CDE uses soft top-k or STE-hard assignments over shared or per-layer codebooks.

RWKV deep embedding. RWKV [5] is an RNN-like alternative to Transformers. Public notes for RWKV-8 "Heron" (2025) describe discrete deep embedding across depth; here it motivates a continuous, vision-specific variant.

3 Method

Notation. Let B be batch size, N tokens per image, d embedding width, and $d_{\mathrm{ff}} \approx 4d$ the MLP hidden size. A pre-norm ViT block yields MLP input $z \in \mathbb{R}^{B\times N\times d}$. We gate either (i) the MLP *output* $y \in \mathbb{R}^{B\times N\times d}$ (*width* mode; $d_g = d$), or (ii) the *expanded hidden* $h \in \mathbb{R}^{B\times N\times d_{\mathrm{ff}}}$ after the first linear and activation (*expand* mode; $d_g = d_{\mathrm{ff}}$).

Codebook and assignment. Let $C \in \mathbb{R}^{K\times d}$ be a codebook and $E \in \mathbb{R}^{K\times d_g}$ a gate matrix, shared across depth unless stated. For token z_{bn}, define cosine logits

$$s_{bnk} = \tau \left\langle \frac{z_{bn}}{\|z_{bn}\|_2}, \frac{c_k}{\|c_k\|_2} \right\rangle. \tag{1}$$

Assignments are either (soft) $\alpha_{bn} = \mathrm{softmax}\big(\text{top-}k(s_{bn\cdot})\big)$ or (hard) $\alpha_{bn} = \mathrm{one\_hot}(\arg\max_k s_{bnk})$ with STE. The gate is

$$g_{bn} = \sum_{k=1}^{K} \alpha_{bnk} E_k \in \mathbb{R}^{d_g}. \tag{2}$$

Apply multiplicative scaling to the chosen MLP tensor (hidden in *expand*, output in *width*):

$$\tilde{y}_{bn} = y_{bn} \odot \left(1 + \lambda\, g_{bn}\right), \quad \lambda \in [0, 1]. \tag{3}$$

The strength λ ramps linearly from $0 \to 1$ during warmup (*identity start*). In the residual block, $x_{l+1} = x_l + \mathrm{DropPath}(\tilde{y})$.

Width vs. expand. *Width* gates d-dimensional outputs; overhead scales with d. *Expand* gates d_{ff}-dimensional hidden activations; overhead scales with $d_{\mathrm{ff}} \approx 4d$ and allows shaping activations before projection back to d.

[1] Code and instructions: github.com/mosure/continuous_deep_embed.

Table 1: Notation summary.

Symbol	Definition
B	batch size
N	tokens per image (e.g., 197 for 14×14 patches plus class token)
d	embedding width
d_{ff}	MLP hidden size ($\approx 4d$)
d_g	gated dimension (d in *width*, d_{ff} in *expand*)
L	number of blocks
K	codebook size
k	mixture size (top-k; k=1 for VQ)
C, E	codebook and gate matrix
τ	temperature for logits
λ	gate strength (ramp $0 \to 1$)

Depth sharing and assignment frequency. By default (C, E) are shared across layers to constrain parameters (*shared-depth*). A per-layer option uses distinct (C_ℓ, E_ℓ) for each block. Independently, logits (1) can be evaluated *once per image* and reused across depth (assign-once) or *per layer* (assign-per-layer). These switches affect both parameters and FLOPs.

4 Complexity

Let k be mixture size (k=1 for VQ) and $d_g \in \{d,\ d_{\text{ff}}\}$.
Assignment. Cosine logits cost NKd, evaluated *once* per image or *per layer*:

$$C_{\text{assign}} = NKd \times \begin{cases} 1 & \text{assign-once} \\ L & \text{assign-per-layer} \end{cases}.$$

Mixing and application (per layer). $C_{\text{mix}} = NLkd_g, \quad C_{\text{apply}} = NLd_g.$
Parameters. Shared-depth: $\#\theta_{\text{CDE}} = Kd + Kd_g$. Per-layer: multiply by L. We report estimated GFLOPs as DeiT-S/224 baseline ($\approx$ 4.6G) plus the terms above (MAC$\rightarrow$FLOP conversion as in standard practice).

5 Training

Identity start. Ramp $\lambda : 0 \to 1$ over T epochs (typically 5–10).
Temperature annealing. Increase τ from softer to sharper values; for VQ, keep $\tau \gtrsim 12$ early.
Mixture size. For ViT-S, $K \in \{256, 512\}$ and $k \in \{4, 8\}$ balance accuracy and overhead.

6 Experiments

6.1 Setup

Backbone: ViT-S/16. **Datasets:** ImageNet-1k [6] (license-respecting preparation) and public subsets (Imagenette/Imagewoof).

Gating modes: width (d_g=d) and expand (d_g=$4d$).

Table 2: ImageNet-1k validation accuracy and compute (placeholders).

Model	Mode	K	top-k	Params (M)	GFLOPs	Top-1 (%)
ViT-S/16 (baseline)	—	—	—	22.1	4.6	**XX.X**
CDE-Soft	width	512	4	22.6	4.8	XX.X
CDE-Soft	expand	512	4	26.3	5.1	XX.X
CDE-VQ	width	512	1	22.6	4.7	XX.X
CDE-VQ	expand	512	1	26.3	4.9	XX.X

Shared-depth unless stated; identical training and augmentation across rows.

Table 3: Ablations on codebook size K, mixture size k, gating dimension d_g, and depth sharing (placeholders).

Variant	Mode	Share	K	top-k	GFLOPs	Top-1 (%)
Soft	width	shared	256	4	4.7	XX.X
Soft	width	shared	512	8	4.9	XX.X
Soft	expand	shared	512	4	5.1	XX.X
VQ	width	shared	512	1	4.7	XX.X
Soft	width	per-layer	512	4	5.2	XX.X

"Share" indicates whether (C, E) are shared across layers. Per-layer increases parameters by $\times L$ and typically uses assign-per-layer logits.

Assignments: soft top-k and hard VQ (STE).

Depth sharing: shared (C, E) across layers; per-layer ablations when noted.

Training: cosine schedule with warmup, AdamW, standard augmentations.

Metrics: ImageNet top-1; GFLOPs estimated as in §4.

6.2 Main Results (placeholders)

6.3 Ablations (placeholders)

7 Implementation and Resources

Code, training harness, dataset preparation scripts, and experiment manifests are available at:

https://github.com/mosure/continuous_deep_embed

The harness logs per-configuration metrics to MLflow, saves *best* and *last* checkpoints for each grid entry, exports CSV and LaTeX manifests, and supports shared or per-layer codebooks with assign-once or assign-per-layer options.

8 Limitations

Assignment scales with NKd; large K or long token sequences raise cost. The present work modulates MLP paths; extensions to attention/value projections are straightforward but not evaluated.

9 Conclusion

Continuous Deep Embed provides a codebook-gated modulation for ViTs inspired by discrete deep embedding in RWKV-8 "Heron." It preserves block structure, adds analyzable overhead, and integrates with standard training. With suitable schedules, CDE matches or slightly improves baseline performance at low overhead.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[2] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 2022.

[3] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

[4] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

[5] Bo Peng et al. RWKV: Reinventing RNNs for the transformer era. arXiv preprint arXiv:2305.13048, 2023.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[7] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

[8] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[9] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.