D-ViCe: A Dual Variance-Enhanced Contrastive Framework for Graph Representation Learning

Anonymous authors
Paper under double-blind review

Abstract

Contrastive learning (CL) has become one of the most powerful paradigms in visual self-supervised learning (SSL), but its extension to non-visual modalities such as graphs and CAD data remains underexplored. Although SSL in graphs is closely related to vision, the transfer is non-trivial in practice because graph and CAD representations lack the regular structure and mature augmentation priors of images. For SSL on graph-based CAD representations, we argue that the design of the contrastive targets and learning dynamics can reduce the reliance on carefully tuned, task-specific augmentations. In this work, we propose a novel **D**ual **V**ari**a**nce-Enhanced **C**ontrastive L**e**arning (**D-ViCe**) framework that employs a dual contrastive mechanism between two target encoders and a shared source encoder. The first target encoder, kept frozen at its random initialization, provides a stationary target for global instance discrimination, while the second, variance-adaptive momentum encoder promotes the refinement of locally invariant representations. Temperature and momentum are central to many CL objectives, yet they are typically fixed or follow predetermined schedules. In contrast, D-ViCe uses training statistics as explicit control signals: the positive-pair probability modulates the contrastive temperature in one branch, and the relative variance of source-teacher similarities governs the target momentum update in the other, thereby balancing global discrimination and local consistency. Experiments on CAD retrieval and graph classification benchmarks show that D-ViCe consistently outperforms state-of-the-art SSL baselines, improving Recall@1 on SolidLetters from 0.515 to 0.798, and yields systematic gains when added to existing graph contrastive methods.

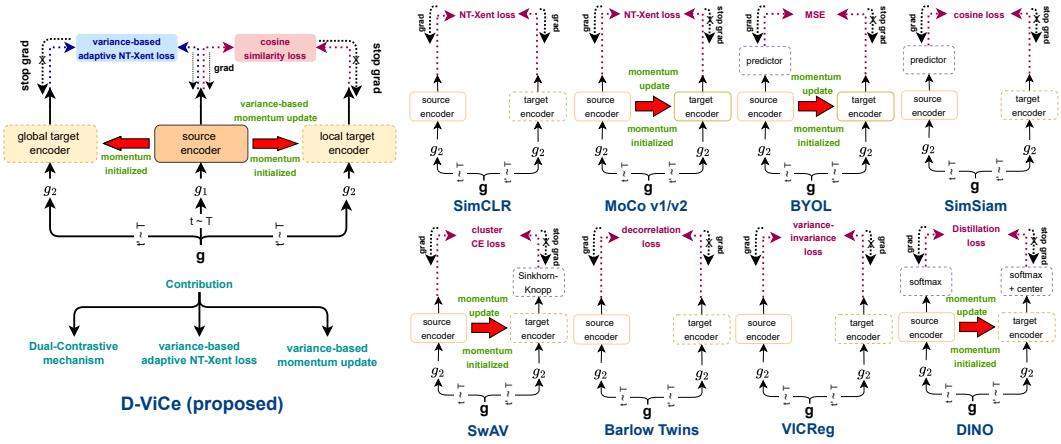

Figure 1: **Proposed D-ViCe framework vs various SOTA architectures.**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Self-supervised learning (SSL) learns transferable representations without large-scale semantic annotations. Contrastive learning (CL) has been particularly successful: it makes the representations of augmented views of the same instance agree while separating those of different instances (He et al., 2020; Misra & Maaten, 2020; Chen et al., 2020; 2021). These objectives, and the hyperparameters that govern them, were developed largely on natural images, whose regular structure supports mature augmentation and optimization protocols. Extending them to graph-structured data is less straightforward, because graph perturbations can alter connectivity and structural relationships, which makes both augmentation design and representation learning strongly domain-dependent (You et al., 2020; Thakoor et al., 2022; You et al., 2021).

This challenge is acute in computer-aided design (CAD). Industrial solids are stored as boundary representations (B-reps), which jointly encode geometry and topology through faces, edges, and their incidence relations (Weiler, 1986). Retrieving related models from large collections enables design reuse, yet labeling such collections is expensive, which makes self-supervised retrieval attractive. Graph encoders such as UV-Net (Jayaraman et al., 2021) represent B-rep structure effectively, but self-supervised learning for CAD remains underexplored, and existing approaches largely adapt objectives developed for images or generic graphs (Jayaraman et al., 2021; Qin et al., 2025). This raises a central question: *how should the targets and the optimization dynamics of contrastive learning be designed when the representation distribution of structured CAD graphs evolves during training?*

Graph SSL methods differ mainly in their augmentation strategies (Veličković et al., 2019; You et al., 2020; 2021; Xu et al., 2021), contrastive mechanisms (Suresh et al., 2021; Xia et al., 2022a; Qiu et al., 2020) and optimization objectives (Zhu et al., 2020; Xia et al., 2022b; Grill et al., 2020; Bardes et al., 2021). Augmentation design is inherently representation-specific: a perturbation that is meaningful in one graph domain may destroy information in another (Jin et al., 2020; Gui et al., 2024). Optimization mechanisms, such as the temperature of contrastive objectives and momentum-updated target encoders, are more broadly transferable, but they are usually governed by fixed values or predetermined schedules. We therefore focus on the *targets and learning dynamics* of the objective rather than on another CAD-specific augmentation family. To ground this choice, we first compare eight SSL methods from four families on the same B-rep encoder (Section 3.1) and find that they transfer very unevenly, which points to three limitations.

Moving targets. Teacher-based methods rely on a target network that evolves during training, which makes the target non-stationary; on B-rep graphs, whose augmentations are far less mature than those for images, this is fragile. *Fixed contrastive scale.* The temperature of InfoNCE or NT-Xent (Oord et al., 2018; Chen et al., 2020) sets the sharpness of the similarity distribution and thus how strongly positive and negative pairs contribute to the gradient, yet it is held constant even though positive-pair confidence changes substantially during training. *Fixed teacher tracking.* Momentum-updated teachers (He et al., 2020; Grill et al., 2020) provide temporally smoothed targets, but their update rate follows a schedule that ignores how well the source and teacher are currently aligned.

We propose **D-ViCe** (**D**ual **V**ari**i**ance-Enhanced **C**ontrastive L**e**arning), a three-encoder SSL framework built on UV-Net that addresses each limitation with one component (Figure 1). The box below summarizes the design.

Moving targets *Can instance discrimination use a target that never drifts and cannot collapse?*	$\Longrightarrow$	**Frozen random teacher** Contrast the source encoder against a fixed, randomly initialised teacher with NT-Xent.
Fixed contrastive scale *Should the sharpness of the contrast respond to how reliably positives are identified?*	$\Longrightarrow$	**Probability-adaptive temperature** Steer the temperature with the running positive-pair probability.
Fixed teacher tracking *Should the teacher follow the source faster when alignment suddenly becomes uneven?*	$\Longrightarrow$	**Variance-adaptive momentum** Modulate the EMA momentum by the relative variance of source-teacher similarities.

A *global* branch performs instance discrimination against the frozen teacher, whose parameters stay fixed throughout training. Its temperature is adjusted from the running positive-pair probability: when confidence falls below a reference level, the controller sharpens the contrastive distribution. A

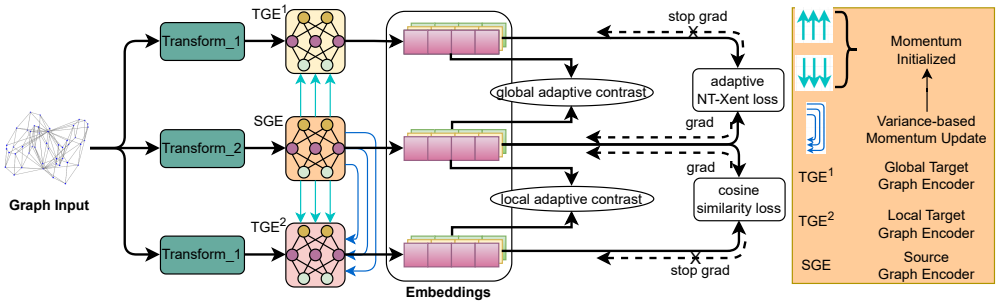

Figure 2: **D-ViCe.** The source encoder E_s is trained against a frozen random teacher E_{t1} (global, confidence-adaptive temperature) and an EMA teacher E_{t2} (local, variance-adaptive momentum), using two augmented views of each B-rep graph.

local branch aligns the source with a second, EMA-updated teacher on corresponding views. The batch variance of their similarities is compared with its running estimate: increased dispersion accelerates teacher adaptation, whereas reduced dispersion yields smoother updates. Neither mechanism adds a representation-level regularizer; both act only on the optimization of the original objectives and are bounded. For the temperature loop, we characterize when the feedback has a stable interior operating point and show that on B-rep graphs it instead settles at its lower bound within the first few dozen steps (Section 2.2). Because the components act on the objective rather than on the encoder or the augmentations, they can also be added to existing graph contrastive methods.

On SolidLetters retrieval, D-ViCe improves Recall@1 from 0.521 for the strongest baseline to 0.798 and Recall@5 from 0.858 to 0.950, consistently across six retrieval back-ends. Added to seven graph contrastive methods on eight TU benchmarks, it improves mean accuracy in all 54 method–dataset pairs, by 0.42 points on average. Our contributions are: ❶ **SSL Benchmark for B-rep CAD.** We compare eight methods from four SSL families on a common UV-Net encoder and six retrieval back-ends and show that the objective, not the retrieval back-end, determines retrieval quality (Section 3.1). ❷ **Dual-Teacher Contrast.** D-ViCe combines global instance discrimination against a frozen random teacher with local alignment to an EMA teacher; the frozen teacher outperforms EMA-updated alternatives (Section 3.2). ❸ **Feedback-Controlled Dynamics.** We adapt the NT-Xent temperature from the running positive-pair probability and the teacher momentum from the relative variance of source–teacher similarities and analyze when the temperature loop has a stable operating point and why it saturates on B-rep graphs (Section 2). ❹ **Empirical Validation.** D-ViCe achieves the best performance among all evaluated SSL methods on SolidLetters under a matched protocol, and its components provide a small but systematic improvement when added to seven graph contrastive methods on eight TU benchmarks (Section 3). We discuss related work in Appendix A.

2 D-ViCe: Dual Variance-Enhanced Contrastive Learning

D-ViCe trains a UV-Net encoder (Jayaraman et al., 2021) with two objectives whose learning dynamics are regulated by feedback from observable training statistics. A *global* branch performs instance discrimination against a frozen teacher and adapts its temperature from the positive-pair probability (assignment *confidence*). A *local* branch aligns the source with an EMA teacher and adapts the teacher's momentum from the relative *variance* of source–teacher similarities. Neither controller adds a loss term; both act only on the optimization dynamics. Section 2.1 describes the architecture, Sections 2.2–2.3 the two branches, and Section 2.4 the full procedure.

2.1 Overview

D-ViCe uses three networks with identical architecture (Fig. 2): a trainable source E_s with parameters $\boldsymbol{\theta}_s$; a *frozen, randomly initialized* teacher E_{t1}; and an EMA teacher E_{t2} with parameters $\boldsymbol{\theta}_{t2}$. Each network is a UV-Net backbone that maps a B-rep face-adjacency graph to a graph embedding $\mathbf{h} \in \mathbb{R}^{128}$, followed by its own projection head ϕ_j and ℓ_2 normalization. Given two independent

augmentations $(\mathcal{G}_1, \mathcal{G}_2)$ of a graph,

$$\begin{aligned} \mathbf{z}_s &= \mathrm{norm}\big(\phi_s(E_s(\mathcal{G}_1))\big), \\ \mathbf{z}_{t1} &= \mathrm{norm}\big(\phi_{t1}(E_{t1}(\mathcal{G}_2))\big), \qquad \mathbf{z}_{t2} = \mathrm{norm}\big(\phi_{t2}(E_{t2}(\mathcal{G}_2))\big). \end{aligned} \tag{1}$$

Only $\boldsymbol{\theta}_s$ receives gradients. Downstream tasks use the pre-projection embedding $\mathbf{h}$ of E_s.

2.2 Global Branch: Confidence-Adaptive Contrast

Objective. For a batch of B graphs, stack $\mathbf{Z} = [\mathbf{Z}_s; \mathbf{Z}_{t1}] \in \mathbb{R}^{2B\times D}$ and let $s_{ij} = \mathbf{z}_i^\top \mathbf{z}_j$. For anchor i with positive i^+ and negatives $\mathcal{N}(i)$ ($|\mathcal{N}(i)| = 2B-2$), the positive-pair probability at step k and the NT-Xent loss (Chen et al., 2020) are

$$p_i(\tau_k) = \frac{\exp(s_{i,i^+}/\tau_k)}{\exp(s_{i,i^+}/\tau_k) + \sum_{j\in\mathcal{N}(i)} \exp(s_{ij}/\tau_k)}, \tag{2}$$

$$\mathcal{L}^{(k)}_{\mathrm{global}} = -\frac{1}{2B}\sum_{i=1}^{2B} \log p_i(\tau_k). \tag{3}$$

Why a frozen random teacher? E_{t1} is never updated after initialization. Contrasting against it asks E_s to identify, among $2B-1$ candidates, the random-feature image of the other view of the same solid. Random networks can provide useful targets: predicting a fixed random target already improves linear-probe accuracy far above a random network (Grill et al., 2020), and distilling from random teachers yields useful representations (Sarnthein et al., 2023). A frozen teacher has two properties we rely on: its parameters never drift, and, because it is not optimized, it cannot collapse during training. Its outputs still vary with batch statistics and dropout in the projection head, so the target is stationary in its parameters rather than exactly deterministic. Table 4 confirms that both matter: letting the global teacher follow the source by EMA lowers R@5 by 4.76 points, and a variance-adaptive update lowers it by 6.07 points.

Temperature response. Let $q_{ij}(\tau)$ be the softmax over candidates $\mathcal{C}(i) = \mathcal{N}(i) \cup \{i^+\}$ and $\tilde{s}_i = \sum_{j\in\mathcal{N}(i)} \frac{q_{ij}}{1-p_i} s_{ij}$ the softmax-weighted mean *negative* similarity. Differentiating Eq. equation 2 gives

$$\frac{\partial \ell_i}{\partial s_{i,i^+}} = -\frac{1-p_i}{\tau}, \qquad \frac{\partial p_i}{\partial \log \tau} = -\frac{p_i(1-p_i)}{\tau}\big(s_{i,i^+} - \tilde{s}_i\big). \tag{4}$$

The first identity shows that the gradient on the positive similarity is governed jointly by τ and p_i. The second shows that whenever the positive is more similar than the weighted negatives, lowering τ raises p_i. Hence p_i is a natural feedback signal for τ.

Controller. We track the batch mean $\bar{p}_k = \frac{1}{2B}\sum_i p_i(\tau_k)$ with an EMA $\hat{p}_k = \lambda \hat{p}_{k-1} + (1-\lambda)\bar{p}_k$ (initialized at $\hat{p}_0 = \bar{p}_0$) and update the log-temperature towards a reference confidence p^*:

$$\boxed{\log \tau_{k+1} = \mathrm{clip}_{[\log \tau_{\min}, \log \tau_{\max}]}\big(\log \tau_k + \kappa\,(\hat{p}_k - p^*)\big).} \tag{5}$$

If confidence is below target, τ decreases and the distribution sharpens; otherwise it softens. The loss at step k always uses τ_k. Algorithm 2 in Appendix D gives the complete procedure for one training step of this branch.

Operating regime. The feedback is corrective only while p^* is attainable. Appendix B characterizes when it is: if every positive is more similar than its softmax-weighted negatives, $\bar{p}$ decreases monotonically in τ, at most one τ^* satisfies $\bar{p}(\tau^*) = p^*$, and the controller converges to it for $\kappa(1-\lambda) < 4\tau_{\min}(1+\lambda)$ (Proposition 1). On B-rep graphs, however, p^* exceeds the attainable positive-pair probability even at $\tau_{\min}$. For the trained model, the batch-mean positive probability at $\tau_{\min}$ is only $\bar{p} \approx 0.003$, $1.6\times$ the chance level $1/(2B-1)$, because source embeddings of different solids remain more similar to each other (0.41) than to their positives (0.39); the monotonicity condition of Proposition 1 holds for only 5.7% of anchors. Since $\hat{p}_k \ll p^*$, each step lowers $\log\tau$ by about $\kappa p^* \approx 0.016$, so τ reaches $\tau_{\min}$ within about 45 steps and stays there. In this regime the controller acts as a short, data-driven decay from τ_0 to $\tau_{\min}$, after which the global branch is NT-Xent at the fixed temperature $\tau_{\min}$; its main effect is to spread the source embeddings apart, while the local branch provides alignment.

2.3 LOCAL BRANCH: VARIANCE-ADAPTIVE MOMENTUM

Objective. The source is aligned with the EMA teacher on corresponding views:

$$\mathcal{L}_{\text{local}}^{(k)} = \frac{1}{B}\sum_{i=1}^{B}\big[2 - 2\,\mathbf{z}_{s,i}^{\top}\,\text{sg}(\mathbf{z}_{t2,i})\big] = \frac{1}{B}\sum_{i=1}^{B}\big\|\mathbf{z}_{s,i} - \text{sg}(\mathbf{z}_{t2,i})\big\|_2^2, \tag{6}$$

where sg is stop-gradient. Unlike BYOL (Grill et al., 2020), there is *no predictor*. On its own, Eq. equation 6 admits the constant (collapsed) solution; in D-ViCe collapse is excluded by the global term, whose negatives include source–source pairs and therefore penalize a constant $\mathbf{z}_s$ (Wang & Isola, 2020). The global term therefore acts as the collapse-preventing component of D-ViCe, and the local term as the alignment component.

EMA teacher. The teacher follows

$$\boldsymbol{\theta}_{t2}^{(k+1)} = m_k\,\boldsymbol{\theta}_{t2}^{(k)} + (1 - m_k)\,\boldsymbol{\theta}_{s}^{(k)}, \tag{7}$$

so each step retains a fraction m_k of the current source–teacher gap, $\|\boldsymbol{\theta}_{t2}^{(k+1)} - \boldsymbol{\theta}_{s}^{(k)}\| = m_k\|\boldsymbol{\theta}_{t2}^{(k)} - \boldsymbol{\theta}_{s}^{(k)}\|$. The momentum m_k therefore directly sets how quickly the teacher tracks the source. Instead of prescribing it by schedule alone, we modulate it by alignment feedback.

Relative-variance feedback. Let $c_{k,i} = \mathbf{z}_{s,i}^{\top}\mathbf{z}_{t2,i}$. We compute its batch variance $v_k = \text{Var}_i(c_{k,i})$, an EMA $\hat{v}_k = \beta\hat{v}_{k-1} + (1-\beta)v_k$ (initialized at $\hat{v}_0 = v_0$), and the ratio

$$r_k = \text{clip}_{[r_{\min}, r_{\max}]}\Big(\frac{v_k}{\hat{v}_k + \epsilon}\Big). \tag{8}$$

The EMA averages over roughly the last $1/(1-\beta) = 50$ batches, so r_k exceeds one when alignment in the current batch is more heterogeneous than in the preceding batches, and falls below one when it is more uniform. It modulates a cosine schedule (Grill et al., 2020):

$$\boxed{m_k = \text{clip}_{[m_{\min},\,m_{\max}]}\big(m_k^{\cos}(2 - r_k)\big), \qquad m_k^{\cos} = 1 - \tfrac{1-m_0}{2}\big[1 + \cos(\pi k/T)\big].} \tag{9}$$

Algorithm 3 in Appendix D details one update of the local teacher.

Rationale. A rise in dispersion ($r_k > 1$) means some samples are now much less aligned with the teacher than others. Our hypothesis is that the teacher has become stale relative to a source that has just moved, so we lower m_k and let the teacher catch up. When dispersion falls ($r_k < 1$), alignment is homogeneous, and we raise m_k to keep the target smooth. Table 3 evaluates this design by comparing VA-EMA with a standard EMA teacher.

Remark 1 (What VA-EMA does and does not do). **(a) Scale invariance.** Rescaling $c_{k,i} \mapsto \alpha c_{k,i}$ leaves r_k unchanged (up to ϵ), so no target variance needs tuning. **(b) Reduction.** If v_k is stationary, then $\hat{v}_k \to v_k$, $r_k \to 1$, and m_k reduces to the clipped cosine schedule; VA-EMA therefore responds to *changes* in dispersion, not to its level. **(c) Range and asymmetry.** Since $m_k^{\cos} \geq m_0$, the source weight $1 - m_k$ ranges over $[1 - m_{\max},\, 1 - m_{\min}] = [10^{-4},\, 0.118]$, up to about $6\times$ the base rate $1 - m_0$. The upward branch saturates quickly: any $r_k \leq 2 - m_{\max}/m_k^{\cos}$ drives m_k to $m_{\max}$. This threshold is 0.980 at the start of training and rises to 1.0001 as $m_k^{\cos} \to 1$, so decreases in dispersion nearly freeze the teacher, whereas increases produce a graded speed-up. For the trained model, we estimate that about two thirds of batches have $m_k = m_{\max}$, i.e. the teacher advances mainly in bursts triggered by rising dispersion. Proofs are given in Appendix C.

2.4 OVERALL OBJECTIVE AND TRAINING

The source minimizes

$$\mathcal{L}_{\text{D-ViCe}} = \mathcal{L}_{\text{global}} + \gamma\,\mathcal{L}_{\text{local}}. \tag{10}$$

E_{t1} stays fixed, and E_{t2} follows Eq. equation 7. The two controllers use disjoint signals and act on disjoint quantities: confidence $\hat{p}_k$ sets the contrastive scale τ_k, and relative dispersion r_k sets the teacher rate m_k. Because the source and the EMA teacher are updated by different rules, the joint dynamics are not gradient descent on a single objective, as is also the case for BYOL (Grill et al., 2020); we therefore analyze the temperature loop in isolation (Appendix B) and evaluate the local loop empirically. Algorithm 1 gives the full procedure. Appendix L discusses the main design choices of D-ViCe, such as why the global teacher is frozen while the local one is updated, together with the evidence behind each.

Algorithm 1: D-ViCe training

Input: Data $\mathcal{D}$; networks E_s, E_{t1}, E_{t2}; γ, τ_0, m_0, p^*, κ, λ, β, clip ranges (Appendix E)
Output: Trained source encoder E_s
Randomly initialize $\boldsymbol{\theta}_s, \boldsymbol{\theta}_{t1}, \boldsymbol{\theta}_{t2}$; set $\boldsymbol{\theta}_{tj} \leftarrow m_0\boldsymbol{\theta}_{tj} + (1-m_0)\boldsymbol{\theta}_s$, $j \in \{1,2\}$
Freeze $\boldsymbol{\theta}_{t1}$; set $\log \tau_0$; mark $\hat{p}, \hat{v}$ uninitialized
for $k = 0, \ldots, T-1$ **do**
  Sample a batch, draw views $\mathcal{G}_1, \mathcal{G}_2$; compute $\mathbf{Z}_s, \mathbf{Z}_{t1}, \mathbf{Z}_{t2}$ by Eq. equation 1 (no gradients through targets)
  $\mathcal{L}_{\text{global}} \leftarrow \text{NTXent}(\mathbf{Z}_s, \mathbf{Z}_{t1}; \tau_k)$; update $\hat{p}_k$ and τ_{k+1} by Eq. equation 5
  $\mathcal{L}_{\text{local}} \leftarrow$ Eq. equation 6; $\mathcal{L} \leftarrow \mathcal{L}_{\text{global}} + \gamma\mathcal{L}_{\text{local}}$
  Compute $v_k, \hat{v}_k, r_k, m_k$ by Eqs. equation 8–equation 9; update $\boldsymbol{\theta}_{t2}$ by Eq. equation 7
  $\boldsymbol{\theta}_s \leftarrow \text{AdamStep}(\boldsymbol{\theta}_s, \nabla_{\boldsymbol{\theta}_s}\mathcal{L})$

Table 1: **CAD retrieval on SolidLetters.** All methods train the same UV-Net encoder with the same data and schedule; only the self-supervised objective differs. **Violet** and blue mark the best and second-best result per column. †BYOL without its predictor head; the standard BYOL reaches R@5 = 0.759 (Cosine). ‡DINO Caron et al. (2021) without a prototype head and with entropy-controlled student and teacher temperatures.

Methods	Cosine			Euclidean			FAISS			HNSW			KNN			Learned		
	R@1	R@5	AUC	R@1	R@5	AUC	R@1	R@5	AUC	R@1	R@5	AUC	R@1	R@5	AUC	R@1	R@5	AUC
BYOL†	0.372	0.780	0.625	0.343	0.762	0.697	0.343	0.762	0.696	0.371	0.778	0.626	0.372	0.780	0.625	0.372	0.780	0.625
SimSiam	0.395	0.814	0.641	0.403	0.813	0.642	0.400	0.811	0.639	0.399	0.818	0.645	0.397	0.814	0.645	0.397	0.816	0.647
BarlowTwins	0.440	0.806	0.656	0.349	0.737	0.760	0.440	0.803	0.657	0.429	0.797	0.649	0.437	0.802	0.654	0.458	0.816	0.670
VICReg	0.475	0.829	0.664	0.441	0.809	0.655	0.469	0.826	0.660	0.463	0.823	0.656	0.469	0.826	0.661	0.488	0.839	0.674
SwAV	0.402	0.827	0.654	0.405	0.825	0.652	0.405	0.825	0.652	0.402	0.826	0.654	0.403	0.827	0.655	0.402	0.827	0.654
DINO‡	0.515	0.844	0.678	0.487	0.831	0.664	0.504	0.840	0.671	0.495	0.837	0.668	0.502	0.841	0.672	0.521	0.851	0.685
MoCo	0.485	0.825	0.631	0.485	0.825	0.631	0.485	0.825	0.631	0.484	0.824	0.632	0.485	0.825	0.631	0.485	0.825	0.631
SimCLR	0.475	0.858	0.655	0.476	0.858	0.649	0.476	0.858	0.649	0.475	0.858	0.655	0.475	0.858	0.655	0.475	0.858	0.655
D-ViCe	**0.798**	**0.950**	**0.809**	**0.798**	**0.948**	**0.836**	**0.798**	**0.948**	**0.836**	**0.798**	**0.949**	**0.810**	**0.798**	**0.950**	**0.809**	**0.798**	**0.950**	**0.809**

2.5 Implementation Details

Backbone and inputs. Each network is a UV-Net (Jayaraman et al., 2021): a 1D CNN encodes each edge's 10-sample UV-grid (XYZ coordinates) into 64-D, a 2D CNN encodes each face's 10×10 UV-grid (XYZ and trimming mask) into 64-D, and a graph network aggregates them into a 128-D embedding $\mathbf{h}$; normals and tangents are not used. The projection head is a three-layer MLP ($128 \to 128 \to 128 \to 64$, BatchNorm and ReLU after the first two layers, dropout 0.3 after the first), followed by ℓ_2 normalization, as in the contrastive setup of UV-Net (Jayaraman et al., 2021). For the TU benchmarks (Morris et al., 2020), the D-ViCe components are added to each base method while its own encoder is kept unchanged (Section 3.1).

Optimization. We train with a batch size of 256 graphs per GPU on two NVIDIA A6000 GPUs, using Adam (Kingma & Ba, 2014) with L_2 weight decay 10^{-4} and a constant learning rate of 10^{-3} for 1000 epochs. The main D-ViCe settings are τ_0=0.1, p^*=0.65, κ=0.025, λ=β=0.98, m_0=0.98 and γ=1; the complete list, including all clipping bounds, is given in Appendix E (Table 6).

Augmentation and evaluation. Augmentations comprise random surface perturbation, edge dropout, and geometric scaling. Retrieval uses $\mathbf{h}$ from E_s with the six retrieval back-ends of Table 1 (Section 3.1). We evaluate on SolidLetters (Jayaraman et al., 2021) and ABC (Koch et al., 2019) for CAD retrieval, and on biochemical and social-network graphs from TU (Morris et al., 2020) for graph classification.

3 Experiments

3.1 Evaluation of D-ViCe

Setup. We compare D-ViCe with eight self-supervised methods from four families: non-contrastive (BYOL (Grill et al., 2020), SimSiam (Chen & He, 2021)), redundancy reduction (Barlow Twins (Zbontar et al., 2021), VICReg (Bardes et al., 2021)), clustering and self-distillation



Figure 3: **Qualitative retrieval with D-ViCe** on SolidLetters (left) and ABC (right). Column 1: query; columns 2–9: top retrieved models.

Table 2: **Graph classification on TU benchmarks** (accuracy in %, SVM with 10-fold cross-validation; mean ± std). Base-method results are published values over 5 runs (10 for AD-GCL); D-ViCe* adds the D-ViCe components to each method, keeping its encoder, augmentations and protocol, and uses 5 random seeds. –: not reported for the base method.

Methods	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B
GraphCL	77.87 ± 0.41	74.39 ± 0.45	78.62 ± 0.40	86.80 ± 1.34	71.36 ± 1.15	89.53 ± 0.84	55.99 ± 0.28	71.14 ± 0.44
D-ViCe*	$\mathbf{78.37 \pm 0.35}$	$\mathbf{74.86 \pm 0.44}$	$\mathbf{79.10 \pm 0.43}$	$\mathbf{87.33 \pm 1.39}$	$\mathbf{71.72 \pm 1.01}$	$\mathbf{90.06 \pm 0.75}$	$\mathbf{56.38 \pm 0.38}$	$\mathbf{71.54 \pm 0.47}$
AD-GCL	75.86 ± 0.62	75.04 ± 0.48	75.73 ± 0.51	88.62 ± 1.27	74.89 ± 0.90	85.52 ± 0.79	53.00 ± 0.82	71.49 ± 0.90
D-ViCe*	$\mathbf{76.33 \pm 0.72}$	$\mathbf{75.44 \pm 0.40}$	$\mathbf{76.09 \pm 0.45}$	$\mathbf{89.13 \pm 1.01}$	$\mathbf{75.31 \pm 0.80}$	$\mathbf{85.82 \pm 0.63}$	$\mathbf{53.33 \pm 0.95}$	$\mathbf{71.96 \pm 0.83}$
JOAOv2	78.36 ± 0.53	74.07 ± 1.10	77.40 ± 1.15	87.67 ± 0.79	69.33 ± 0.34	86.42 ± 1.45	56.03 ± 0.27	70.83 ± 0.25
D-ViCe*	$\mathbf{78.69 \pm 0.46}$	$\mathbf{74.42 \pm 0.95}$	$\mathbf{77.83 \pm 0.99}$	$\mathbf{87.79 \pm 0.83}$	$\mathbf{69.57 \pm 0.31}$	$\mathbf{86.55 \pm 1.34}$	$\mathbf{56.41 \pm 0.35}$	$\mathbf{71.20 \pm 0.24}$
RGCL	78.14 ± 1.08	75.03 ± 0.43	78.86 ± 0.48	87.66 ± 1.01	70.92 ± 0.65	90.34 ± 0.58	56.38 ± 0.40	71.85 ± 0.84
D-ViCe*	$\mathbf{79.04 \pm 0.99}$	$\mathbf{75.69 \pm 0.35}$	$\mathbf{79.33 \pm 0.59}$	$\mathbf{88.27 \pm 1.11}$	$\mathbf{71.12 \pm 0.49}$	$\mathbf{91.19 \pm 0.50}$	$\mathbf{57.07 \pm 0.36}$	$\mathbf{72.68 \pm 0.89}$
AutoGCL	82.00 ± 0.29	75.80 ± 0.36	77.57 ± 0.60	88.64 ± 1.08	70.12 ± 0.68	88.58 ± 1.49	56.75 ± 0.18	73.30 ± 0.40
D-ViCe*	$\mathbf{82.11 \pm 0.25}$	$\mathbf{76.19 \pm 0.29}$	$\mathbf{77.91 \pm 0.71}$	$\mathbf{88.99 \pm 1.01}$	$\mathbf{70.34 \pm 0.76}$	$\mathbf{88.78 \pm 1.39}$	$\mathbf{56.89 \pm 0.23}$	$\mathbf{73.64 \pm 0.35}$
GCS	77.37 ± 0.30	75.02 ± 0.39	77.22 ± 0.30	90.45 ± 0.81	75.56 ± 0.41	92.98 ± 0.28	57.04 ± 0.49	73.43 ± 0.38
D-ViCe*	$\mathbf{77.95 \pm 0.33}$	$\mathbf{75.65 \pm 0.49}$	$\mathbf{77.76 \pm 0.25}$	$\mathbf{91.11 \pm 0.61}$	$\mathbf{76.01 \pm 0.42}$	$\mathbf{93.45 \pm 0.29}$	$\mathbf{57.39 \pm 0.36}$	$\mathbf{73.73 \pm 0.29}$
TopoGCL	81.30 ± 0.27	77.30 ± 0.89	79.15 ± 0.35	90.09 ± 0.93	–	90.40 ± 0.53	–	74.67 ± 0.32
D-ViCe*	$\mathbf{81.69 \pm 0.31}$	$\mathbf{77.71 \pm 0.92}$	$\mathbf{79.31 \pm 0.39}$	$\mathbf{90.66 \pm 0.82}$	–	$\mathbf{90.94 \pm 0.61}$	–	$\mathbf{74.91 \pm 0.24}$

(SwAV (Caron et al., 2020), DINO (Caron et al., 2021)), and InfoNCE-based contrastive learning (MoCo (He et al., 2020), SimCLR (Chen et al., 2020)); SimCLR is the objective of the original UV-Net (Jayaraman et al., 2021). All methods use the same UV-Net backbone, data and training schedule; their objectives and associated heads follow the adaptations noted in Table 1. Test solids are used as queries against a gallery of training solids, and a retrieved model is correct if it has the query's letter class. We evaluate the learned embeddings with six retrieval back-ends (Cosine, Euclidean, FAISS, HNSW, KNN and Learned in Table 1); FAISS uses an exact flat index with Euclidean distance, and HNSW an approximate graph index with cosine distance (M=16, $ef_{\text{construction}}$=100, ef=20). KNN is an exact cosine top-k search, and Learned applies a linear projection before cosine similarity; in our evaluation this projection is the identity. We report Recall@1 (R@1), the fraction of queries whose nearest neighbor is correct, Recall@5 (R@5), and AUC, the area under the ROC curve of the top-1 similarity score as a predictor of whether the top-1 retrieval is correct.

CAD retrieval. Table 1 shows that D-ViCe is best on all 18 metric–back-end combinations. The largest gain is in R@1: D-ViCe reaches an R@1 of 0.798 with every back-end, compared with 0.487–0.521 for the strongest baseline, DINO, an absolute gain of 28–31 points. At R@5 the margin over the strongest baseline, SimCLR, is about 9 points (0.950 vs. 0.858). AUC improves on every back-end as well, for example from 0.678 (DINO) to 0.809 with cosine similarity and from 0.760 (Barlow Twins) to 0.836 with Euclidean distance. The large R@1 gain indicates that D-ViCe does not merely place the correct shape somewhere among the top candidates: it places it first four times out of five. Beyond recall, D-ViCe also places far more correct models among the ten nearest neighbors: its precision@10 over all 9,728 test queries is 73.1%, compared with at most 41.8% for any baseline (Appendix H). The t-SNE projections in Appendix I (Fig. 6) show the same trend qualitatively: D-ViCe forms compact, well-separated class clusters, whereas the baselines show overlapping or diffuse class groups.

Comparison among baselines. The ranking of the baselines depends on the metric. SimCLR achieves the second-best R@5 (0.858) but only a mid-range R@1 (0.475), whereas DINO has the second-best R@1 and, on most back-ends, the second-best AUC, but a lower R@5. High top-5 recall therefore does not imply precise top-1 retrieval, which is why we report all three metrics.

The two non-contrastive methods, BYOL and SimSiam, perform worst overall. A query-by-query comparison with the UV-Net SimCLR baseline in Appendix J (Fig. 7) shows that the baseline's errors are mostly shapes that resemble the query locally but differ in global structure, which D-ViCe distinguishes correctly.

Robustness to the retrieval back-end. The choice of back-end matters little: for D-ViCe, the metrics vary by at most 0.002 in R@1 and R@5 and by 0.027 in AUC across the six back-ends. Differences between methods therefore stem from the learned representations rather than from a particular distance or index structure. Appendix K (Fig. 8) confirms this visually: the top-21 retrievals of D-ViCe are nearly identical under k-NN, cosine similarity, FAISS and HNSW.

Model-agnostic evaluation on graph classification. To test whether D-ViCe's components transfer beyond CAD data, we add them to seven graph contrastive learning methods: GraphCL (You et al., 2020), AD-GCL (Suresh et al., 2021), JOAOv2 (You et al., 2021), RGCL (Li et al., 2022), AutoGCL (Yin et al., 2022), GCS (Wei et al., 2023) and TopoGCL (Chen et al., 2024). For each method, D-ViCe* replaces its contrastive loss with VA-NTXent and adds the VA-EMA local branch, while the encoder, augmentations and evaluation protocol of the original method are kept unchanged. Table 2 reports accuracy on four biochemical (NCI1, PROTEINS, DD, MUTAG) and four social-network (COLLAB, RDT-B, RDT-M5K, IMDB-B) benchmarks, following the standard protocol of an SVM classifier with 10-fold cross-validation on the frozen graph embeddings (You et al., 2020).

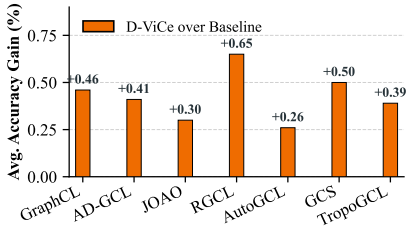

Figure 4: **Average accuracy gain** of D-ViCe* over each base method, averaged over the TU datasets (in %).

Results. D-ViCe* improves the mean accuracy in all 54 method–dataset pairs. Fig. 4 summarizes the average gain per method, which ranges from $+0.26$ points (AutoGCL) to $+0.65$ points (RGCL), with an overall mean of $+0.42$ points. The gains are therefore consistent in direction but modest in size: in 16 of the 54 pairs the gain exceeds the larger of the two standard deviations, and in the remaining pairs it lies within run-to-run variation. The improvement is not tied to one type of method, appearing for augmentation-based (GraphCL, JOAOv2, AD-GCL, AutoGCL), rationale-based (RGCL), saliency-based (GCS) and topology-aware (TopoGCL) approaches alike. D-ViCe can thus be added to existing graph contrastive methods as a plug-in, without changing their encoder or pretext task, and provides a small but systematic benefit.

3.2 Ablation Study

Protocol. All ablations are run on SolidLetters and report recall@5 (R@5) for cosine-similarity retrieval on the 128-D embedding $\mathbf{h}$ of the source encoder. Every configuration is trained with the same schedule, and only the listed factor is changed; all other hyperparameters are those of Table 6. Each configuration is a single run; the empirical run-to-run spread, estimated in Appendix G.3, is about 0.9 points.

Contribution of each component. Table 3 adds the components of D-ViCe one at a time to our reproduction of the UV-Net contrastive baseline (Jayaraman et al., 2021), which is trained with the standard NT-Xent loss and reaches an R@5 of 0.8578. EMA denotes the cosine schedule $m_k^{\cos}$ of Eq. equation 9 without variance feedback. In total, D-ViCe improves R@5 by 9.18 points, to 0.9496. *(i) Global objective.* Replacing NT-Xent with VA-NTXent, which contrasts against the frozen teacher of Section 2.2 at the low temperature its controller settles at, gives the largest single gain ($+6.66$). *(ii) Local branch.* Adding the local branch with a standard EMA teacher improves both global objectives, by $+4.24$ points for NT-Xent but only $+1.82$ points for VA-NTXent, so the two branches are complementary but partly overlapping. *(iii) Variance-adaptive momentum.* Replacing the standard EMA with VA-EMA gives the best result, 0.9496 ($+0.70$), although this increment lies within the run-to-run spread.

Table 3: **Component ablation** (UV-Net backbone). Δ: gain over the NT-Xent baseline.

Global	Local	R@5	Δ
NT-Xent (UVNet)	–	0.8578	–
VA-NTXent	–	0.9244	$+6.66$
NT-Xent	EMA	0.9002	$+4.24$
VA-NTXent	EMA	0.9426	$+8.48$
VA-NTXent	VA-EMA	**0.9496**	$\mathbf{+9.18}$

Frozen vs. updated global teacher. Table 4 isolates the update rule of the global teacher E_{t1} with all other settings fixed. Relative to the frozen random teacher, updating E_{t1} as an exponential moving average of the source at every step (base momentum m_0=0.98 with a cosine schedule) lowers R@5 by 4.76 points, and a variance-adaptive update by 6.07 points. In our ablations, both EMA-based updates substantially degrade retrieval relative to the frozen teacher: for instance discrimination, a parameter-frozen target provides a better reference in our setting than one that tracks the source. The same variance-adaptive update that harms the global teacher benefits the local teacher (Table 3), so the two branches need different kinds of targets.

Table 4: **Update setting of the global teacher.** Only the update of E_{t1} varies; Δ: change w.r.t. the frozen teacher.

Global teacher E_{t1}	**R@5**	**Δ**
Frozen, random (ours)	**0.9496**	–
EMA, every step	0.9020	−4.76
Variance-adaptive EMA	0.8889	−6.07

Weight of the local objective. Table 5 varies γ in Eq. equation 10. With $\gamma = 0$ (global branch only), D-ViCe already exceeds the SimCLR baseline (Jayaraman et al., 2021) by 6.66 points. R@5 increases monotonically up to $\gamma = 1.0$ (0.9358, 0.9441 and 0.9496 for $\gamma = 0.5$, 0.9 and 1.0) and decreases beyond it (0.9452 and 0.9398 for $\gamma = 1.2$ and 1.4), so the optimum lies at $\gamma \approx 1$, where both objectives are weighted equally; the drop at $\gamma = 1.4$ (0.98 points) exceeds the run-to-run spread. Weighting the alignment term more heavily than the discriminative global term therefore reduces retrieval quality, consistent with the complementary roles of the two objectives (Section 2.3). Only E_s receives gradients and is deployed, so the model used at inference is identical in size and speed to the baseline; the additional cost is confined to training, where each target network adds one forward pass per step and no backward pass.

Table 5: **Weight of the local objective γ.** Params: all networks held in memory during training.

Setting	**Params (M)**	**R@5**
SimCLR	1.2	0.8578
$\gamma = 0$ (global-only)	1.2	0.9244
$\gamma = 0.5$	2.3	0.9358
$\gamma = 0.9$	2.3	0.9441
$\gamma = 1.0$	2.3	**0.9496**
$\gamma = 1.2$	2.3	0.9452
$\gamma = 1.4$	2.3	0.9398

Summary. Most of D-ViCe's gain comes from the global branch, whose frozen teacher is essential; because its temperature controller settles at $\tau_{\min}$, this branch operates as NT-Xent against a parameter-frozen random target at a low temperature. The local branch adds 1.82 points, and VA-EMA another 0.70. The two objectives work best when weighted equally ($\gamma \approx 1$). Appendix G completes the study with the sensitivity of both controllers, the run-to-run spread and the batch size; for most controller hyperparameters, we observe no effect distinguishable from this spread.

4 Conclusion and Future Work

We presented **D-ViCe** (Dual Variance-Enhanced Contrastive Learning), a self-supervised framework for B-rep CAD and graph representation learning that combines global instance discrimination against a frozen random teacher with local alignment to an EMA teacher, and regulates the learning dynamics of both branches with bounded feedback controllers: the contrastive temperature from the positive-pair probability and the teacher momentum from the relative variance of source–teacher similarities. On SolidLetters retrieval, D-ViCe outperforms eight self-supervised baselines from four families on all 18 metric–back-end combinations, raising Recall@1 from 0.521 to 0.798 and Recall@5 from 0.858 to 0.950. Added to seven graph contrastive methods, its components improve accuracy in all 54 method–dataset pairs on eight TU benchmarks, by 0.42 points on average. Our ablations show that most of the gain comes from the global branch, and in particular from its frozen teacher: letting the global target follow the source costs up to 6 points, while the same adaptive update benefits the local teacher, so the two branches require different kinds of targets. They also show that the temperature controller settles at its lower bound on B-rep graphs, because the reference confidence is not attainable there (Appendix M). Future work will follow three directions. First, we will define the temperature reference relative to the chance level of the contrastive task, so that the controller remains in its corrective regime, and test whether a continuously adapting temperature improves over a fixed one. Second, we will study augmentation strategies designed jointly with D-ViCe's objectives and evaluate the framework on further CAD corpora and encoders, including graph transformers that capture long-range structural dependencies. Third, we plan to extend D-ViCe to hypergraph representations, to model higher-order relations between faces and edges and to broaden its applicability beyond pairwise graph structure.

AI USE STATEMENT

We used large language model (LLM) based tools as a writing and review aid: to refine grammar and phrasing, restructure and condense sections, draft figure and table captions, suggest candidate related work (each verified by the authors against the original publications), and check consistency between reported values and our result files. LLM tools also assisted in drafting and checking parts of the mathematical exposition and of the scripts used for checkpoint diagnostics; all such contributions were independently verified by the authors, and every reported number was produced by running our own code on our own trained models. The research idea, the design of D-ViCe, the experiments, and all measurements are the authors' own, and we take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

D-ViCe is fully specified in Section 2: the objectives in Eqs. equation 2–equation 6, the two controllers in Eqs. equation 5 and equation 9, and the complete training procedure in Algorithm 1. The architecture, optimizer and training settings are given in Section 2.5, and all hyperparameters are listed in Table 6. The analysis of the temperature controller and all proofs appear in Appendices B and C, and the evaluation protocols for retrieval, graph classification and the ablations are stated in Sections 3.1 and 3.2. We use only publicly available datasets (SolidLetters, ABC and the TU benchmarks) with their standard splits. Source code, configurations and the scripts used to produce all reported results will be released; an anonymized version is provided as supplementary material.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9640–9649, 2021.

Yuzhou Chen, Jose Frias, and Yulia R Gel. Topogcl: Topological graph contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11453–11461, 2024.

Julien Denize, Jaonary Rabarisoa, Astrid Orcesi, Romain Hérault, and Stéphane Canu. Similarity contrastive estimation for self-supervised soft contrastive learning. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2705–2715. IEEE, 2023.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G Lambourne, Karl DD Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. Uv-net: Learning from boundary representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11703–11712, 2021.

Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9601–9611, 2019.

Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. *arXiv preprint arXiv:2303.13664*, 2023.

Joseph G Lambourne, Karl DD Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. *arXiv preprint arXiv:2104.00706*, 2021.

Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In *International conference on machine learning*, pp. 13052–13065. PMLR, 2022.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6707–6717, 2020.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Feiwei Qin, Liangzhe Zhu, Zijian Xu, Meie Fang, and Ping Li. Cadgcl: unsupervised retrieval of cad models via boundary representations. *The Visual Computer*, 41(9):6493–6505, 2025.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Zi-Hao Qiu, Quanqi Hu, Zhuoning Yuan, Denny Zhou, Lijun Zhang, and Tianbao Yang. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. *arXiv preprint arXiv:2305.11965*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Felix Sarnthein, Gregor Bachmann, Sotiris Anagnostidis, and Thomas Hofmann. Random teachers are good teachers. In *International Conference on Machine Learning*, pp. 30022–30041. PMLR, 2023.

Aravind Sankar Suresh, Soumya Chakrabarti, Karthik Subbian, and Hari Sundaram. Adversarial graph contrastive learning with information regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.09968. arXiv:2106.09968.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2102.06514. arXiv:2102.06514.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rklz9iAcKQ.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Chunyu Wei, Yu Wang, Bing Bai, Kai Ni, David Brady, and Lu Fang. Boosting graph contrastive learning via graph contrastive saliency. In *International conference on machine learning*, pp. 36839–36855. PMLR, 2023.

Kevin J Weiler. *Topological structures for geometric modeling (Boundary representation, manifold, radial edge structure)*. Rensselaer Polytechnic Institute, 1986.

Jun Xia, Yuxiao Zhang, Yang Xu, Le Song, and Carl Yang. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *International Conference on Learning Representations (ICLR)*, 2022a. URL https://openreview.net/forum?id=we0vEiiWUpE.

Jun Xia, Yuxiao Zhang, Yang Xu, and Carl Yang. Progcl: Rethinking hard negative mining in graph contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b. URL https://arxiv.org/abs/2202.08670.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11548–11558. PMLR, 2021. URL http://proceedings.mlr.press/v139/xu21g/xu21g.pdf.

Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8892–8900, 2022.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, pp. 5812–5823. Curran Associates, Inc., 2020. URL https://github.com/Shen-Lab/GraphCL.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12121–12132. PMLR, 2021. URL http://proceedings.mlr.press/v139/you21a/you21a.html.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 13387–13398. PMLR, 2020. URL https://proceedings.mlr.press/v119/zhu20a.html.

APPENDIX OVERVIEW

This appendix provides background, theoretical analysis, implementation details, and additional experimental results for D-ViCe (Dual Variance-Enhanced Contrastive Learning). Appendix A reviews related work. Appendices B and C analyze the temperature controller and prove the properties of the momentum controller stated in Remark 1, while Appendix D provides pseudocode for both controllers. Appendix E lists all hyperparameters. Section F evaluates a reduced D-ViCe configuration on ImageNet-1K against representative contrastive and non-contrastive SSL methods under a matched ResNet-50 linear-evaluation protocol. Appendix G then completes the ablation study with the sensitivity of both controllers, the run-to-run spread, and the effect of batch size. Appendices H–K provide further retrieval analyses, including precision@10 over all test queries, t-SNE projections of all nine methods, a query-by-query comparison with the UV-Net baseline, and retrieval consistency across different implementations (generated through our developed GUI). Appendix L addresses common questions regarding D-ViCe's design, and Appendix M discusses its limitations. All CAD retrieval experiments use SolidLetters (Jayaraman et al., 2021). Code and additional visualizations are available in our anonymous repository[1]; the full code and checkpoints will be released upon publication.

A RELATED WORK

Self-supervised representation learning. Most self-supervised methods learn by making the embeddings of two views of the same input agree, and they differ in how they stop the embeddings from collapsing to a constant. Contrastive methods such as CPC, SimCLR and MoCo push apart the views of different inputs with an InfoNCE-type loss (Oord et al., 2018; Chen et al., 2020; He et al., 2020). BYOL and SimSiam need no negatives and instead rely on a predictor and a stop-gradient (Grill et al., 2020; Chen & He, 2021), Barlow Twins and VICReg regularize the statistics of the embedding matrix (Zbontar et al., 2021; Bardes et al., 2021), and SwAV and DINO assign views to learned prototypes or distil a teacher's output distribution (Caron et al., 2020; 2021). We compare against all four families on B-rep data in Section 3.1. D-ViCe itself is contrastive: its global branch uses NT-Xent, and its local branch adds an alignment term to a second teacher.

Temperature in contrastive learning. The temperature decides how sharply the loss concentrates on hard negatives, and Wang & Isola (2020) relate it to the balance between alignment and uniformity of the embedding. It is usually fixed by hand. A few works make it vary: CLIP learns it by gradient descent (Radford et al., 2021), Kukleva et al. (2023) follow a cosine schedule to help on long-tailed data, and Qiu et al. (2023) give each anchor its own temperature. Our controller instead adjusts a single temperature from the positive-pair probability observed during training. On B-rep graphs this probability stays far below the reference value, so in practice the controller reaches its lower bound early and stays there (Section 2.2); we discuss this openly in Appendix M.

Momentum and frozen teachers. MoCo and BYOL compute their targets with a teacher whose weights follow an exponential moving average of the student (He et al., 2020; Grill et al., 2020), and SCE combines such a teacher with a soft contrastive objective (Denize et al., 2023). The momentum is either constant or follows a cosine schedule; in D-ViCe's local branch it also reacts to how unevenly the batch is aligned with the teacher. For the global branch we go further and do not update the teacher at all. This is less odd than it sounds: BYOL already reports that predicting the output of a fixed random network yields features much better than random ones (Grill et al., 2020), and Sarnthein et al. (2023) show that distilling from a random teacher can produce useful representations. We use such a teacher as a contrastive reference and find that it outperforms EMA-updated alternatives (Table 4).

Graph contrastive learning. On graphs, GraphCL established the contrastive recipe of two augmented views and an NT-Xent loss (You et al., 2020). Much of the follow-up work concerns the views: JOAO selects augmentations automatically (You et al., 2021), AD-GCL learns them adversarially (Suresh et al., 2021), AutoGCL learns view generators (Yin et al., 2022), RGCL keeps the rationale of a graph intact (Li et al., 2022), GCS preserves salient substructures (Wei et al., 2023), and TopoGCL adds topological views (Chen et al., 2024). Because D-ViCe changes the objective

[1] https://github.com/anonymous0610/ICLR2027

and the teachers rather than the views, it can be added to each of these methods, which is how we use it in Table 2.

Learning on B-rep CAD models. UV-Net encodes a B-rep as a face-adjacency graph whose nodes and edges carry sampled UV-grids of the surfaces and curves, and it introduced SolidLetters together with a contrastive baseline for self-supervised retrieval (Jayaraman et al., 2021). BRepNet operates on the oriented topology of the B-rep for supervised segmentation (Lambourne et al., 2021), and CAD-GCL studies graph contrastive learning for CAD models (Qin et al., 2025). Self-supervised work in this area still largely reuses a single objective taken from vision. We keep the UV-Net encoder fixed and study which objective and teacher design suit B-rep data best.

B Analysis of the Temperature Controller

This section analyzes the temperature loop of Eq. equation 5 with the confidence EMA $\hat{p}_k = \lambda\hat{p}_{k-1} + (1-\lambda)\bar{p}_k$. It states when the loop has a stable interior operating point and explains why, on B-rep graphs, it settles at $\tau_{\min}$ instead.

Proposition 1 (Monotone response and controller stability)**.** *Hold the representations fixed and write $\bar{p}(\tau)$ for the batch-mean positive probability.* ***(i)*** *If $s_{i,i^+} > \tilde{s}_i$ for every anchor, with $\tilde{s}_i$ the softmax-weighted mean negative similarity of Eq. equation 4, $\bar{p}$ is strictly decreasing in $\log\tau$; hence at most one $\tau^* \in [\tau_{\min}, \tau_{\max}]$ satisfies $\bar{p}(\tau^*) = p^*$.* ***(ii)*** *Let $g = |\mathrm{d}\bar{p}/\mathrm{d}\log\tau|$ at an interior τ^* and $e_k = \log\tau_k - \log\tau^*$. Linearized, the coupled EMA controller obeys*

$$e_{k+1} = \big(1 + \lambda - \kappa(1-\lambda)g\big)e_k - \lambda e_{k-1}, \tag{11}$$

which is asymptotically stable if and only if $\kappa(1-\lambda)g < 2(1+\lambda)$. Because $g \le 1/(2\tau_{\min})$, the condition $\kappa(1-\lambda) < 4\tau_{\min}(1+\lambda)$ is sufficient.

With our settings ($\kappa = 0.025$, $\lambda = 0.98$, $\tau_{\min} = 0.05$), $\kappa(1-\lambda) = 5\times10^{-4}$ against a bound of 0.396. *When an interior τ^* exists*, the roots of Eq. equation 11 are complex whenever $\kappa(1-\lambda)g > (1-\sqrt{\lambda})^2 \approx 10^{-4}$, and τ_k approaches τ^* through a damped oscillation with per-step contraction $\sqrt{\lambda} \approx 0.990$, i.e. a time constant of roughly 100 steps. On B-rep graphs, however, p^* exceeds the attainable positive-pair probability even at $\tau_{\min}$; hence no interior operating point exists and the controller saturates at $\tau_{\min}$ (Section 2.2).

Proof. Throughout, the representations (hence all s_{ij}) are held fixed and only $u = \log\tau$ varies.

Derivative. Write $Z_i = \sum_{j\in\mathcal{C}(i)} \exp(s_{ij}e^{-u})$ and $q_{ij} = \exp(s_{ij}e^{-u})/Z_i$, so that $p_i = q_{i,i^+}$. Since $\mathrm{d}(e^{-u})/\mathrm{d}u = -1/\tau$,

$$\frac{\mathrm{d}\log p_i}{\mathrm{d}u} = -\frac{s_{i,i^+}}{\tau} + \sum_{j\in\mathcal{C}(i)} q_{ij}\frac{s_{ij}}{\tau} = -\frac{1}{\tau}\big(s_{i,i^+} - \bar{s}_i\big), \qquad \bar{s}_i = \sum_{j\in\mathcal{C}(i)} q_{ij}s_{ij}.$$

Splitting $\bar{s}_i$ into its positive and negative parts gives $\bar{s}_i = p_i s_{i,i^+} + (1-p_i)\tilde{s}_i$ with $\tilde{s}_i = \sum_{j\in\mathcal{N}(i)} \frac{q_{ij}}{1-p_i}s_{ij}$, hence $s_{i,i^+} - \bar{s}_i = (1-p_i)(s_{i,i^+} - \tilde{s}_i)$ and

$$\frac{\partial p_i}{\partial\log\tau} = p_i\frac{\mathrm{d}\log p_i}{\mathrm{d}u} = -\frac{p_i(1-p_i)}{\tau}\big(s_{i,i^+} - \tilde{s}_i\big),$$

which is Eq. equation 4. The identity $\partial\ell_i/\partial s_{i,i^+} = -(1-p_i)/\tau$ follows in the same way from $\ell_i = -\log p_i$.

Part (i). If $s_{i,i^+} > \tilde{s}_i$, then $\partial p_i/\partial u < 0$ because $p_i \in (0,1)$ and $\tau > 0$. A mean of strictly decreasing functions is strictly decreasing, so $\bar{p}(u)$ is strictly decreasing and continuous, and $\bar{p}(u) = p^*$ has at most one solution on $[\log\tau_{\min}, \log\tau_{\max}]$. A solution exists if $\bar{p}(\tau_{\max}) \le p^* \le \bar{p}(\tau_{\min})$; otherwise the clip in Eq. equation 5 holds τ at the corresponding bound.

Bound on g. Cosine similarities lie in $[-1,1]$ and $\tilde{s}_i$ is a convex combination of them, so $0 < s_{i,i^+} - \tilde{s}_i \le 2$. With $p_i(1-p_i) \le \frac{1}{4}$ this gives $|\partial p_i/\partial u| \le 1/(2\tau)$ for every anchor, hence $g \le 1/(2\tau^*) \le 1/(2\tau_{\min})$.

Part (ii). At every training step the implementation first sets $\hat{p}_k = \lambda \hat{p}_{k-1} + (1-\lambda)\bar{p}(u_k)$ and then $u_{k+1} = u_k + \kappa(\hat{p}_k - p^*)$. Let u^* be an interior fixed point, $e_k = u_k - u^*$ and $y_k = \hat{p}_k - p^*$, and linearize $\bar{p}(u_k) - p^* \approx -g\, e_k$. Then

$$y_k = \lambda y_{k-1} - (1-\lambda) g\, e_k, \qquad e_{k+1} = e_k + \kappa y_k.$$

The second relation at step $k-1$ gives $y_{k-1} = (e_k - e_{k-1})/\kappa$; substituting into the first and then into the second yields

$$e_{k+1} = \big(1 + \lambda - a\big) e_k - \lambda e_{k-1}, \qquad a := \kappa(1-\lambda) g,$$

i.e. Eq. equation 11, with characteristic polynomial $P(z) = z^2 - (1+\lambda-a)z + \lambda$. By the Jury criterion, both roots lie strictly inside the unit circle if and only if $|\lambda| < 1$, $P(1) = a > 0$ and $P(-1) = 2(1+\lambda) - a > 0$. The first two hold for $\lambda \in [0, 1)$ and $g > 0$, so stability is equivalent to $a < 2(1+\lambda)$, and $g \le 1/(2\tau_{\min})$ gives the sufficient condition $\kappa(1-\lambda) < 4\tau_{\min}(1+\lambda)$. □

Transient. The roots of P are complex if and only if $(1+\lambda-a)^2 < 4\lambda$, i.e. for $(1-\sqrt{\lambda})^2 < a < (1+\sqrt{\lambda})^2$. In that regime both roots have modulus $\sqrt{\lambda}$, so the error contracts by $\sqrt{\lambda}$ per step. For $\lambda = 0.98$, $(1-\sqrt{\lambda})^2 \approx 1.01 \times 10^{-4}$, $\sqrt{\lambda} \approx 0.990$, and the envelope time constant is $-1/\log\sqrt{\lambda} \approx 99$ steps; with $\kappa(1-\lambda) = 5 \times 10^{-4}$, oscillation occurs whenever $g > 0.2$.

Numerical check. On synthetic unit-norm embeddings ($B = 256$, $D = 128$) with frozen representations, the closed-form derivative matches central finite differences to six decimals. When p^* is attainable, the exact update order converges to the unique τ^* with $\bar{p}(\tau^*) = p^*$ and shows the predicted damped oscillation with modulus $\sqrt{\lambda}$. When positives are too hard for p^* to be reached, τ settles at $\tau_{\min}$, as the clipping in Eq. equation 5 implies; this is the regime observed for the trained D-ViCe model (Section 2.2).

C Proof of Remark 1

Proof. **(a)** Under $c \mapsto \alpha c$, both v_k and $\hat{v}_k$ scale by α^2, so their ratio is unchanged when $\epsilon \ll \hat{v}_k$. **(b)** If $v_k \equiv v$ from initialization onward, then $\hat{v}_k = v$ for all k, so $r_k = v/(v+\epsilon) \approx 1$ and $m_k = \mathrm{clip}(m_k^{\cos})$. **(c)** Since $\hat{v}_k$ includes the current term, $r_k = v_k/(\beta\hat{v}_{k-1} + (1-\beta)v_k) \le 1/(1-\beta)$ before clipping. From $m_k^{\cos} \ge m_0$ and $r_k \in [r_{\min}, r_{\max}]$, the pre-clip value lies in $[0,\, 1.5\, m_k^{\cos}]$, and after clipping $1 - m_k \in [1 - m_{\max}, 1 - m_{\min}]$. Finally, $m_k^{\cos}(2 - r_k) \ge m_{\max}$ if and only if $r_k \le 2 - m_{\max}/m_k^{\cos}$. □

D Pseudocode of the Controllers

Algorithms 2 and 3 give one training step of the two controllers of D-ViCe, in the order in which they are executed within Algorithm 1. Both keep a running estimate that is initialized from the first batch. In our implementation the temperature controller is also updated on validation batches; because τ remains at $\tau_{\min}$ throughout training (Section 2.2), this does not affect the reported results, and the released code restricts both controllers to training steps.

E Hyperparameters

Table 6 lists all hyperparameters introduced by D-ViCe, grouped by the component they belong to; the same values are used in every CAD experiment, and the optimizer, batch size and schedule are given in Section 2.5. Of these, only the controller parameters shape the learning dynamics: the reference confidence p^*, the gain κ and the EMA factor λ of the temperature loop (Eq. equation 5), and the base momentum m_0 and EMA factor β of the momentum loop (Eqs. equation 8–equation 9). The remaining entries are the initial temperature τ_0 and numerical safeguards: the clipping bounds of τ, r_k and m_k, and the stability constant ϵ. Both EMA factors are set to 0.98, so $\hat{p}_k$ and $\hat{v}_k$ average over roughly the last $1/(1-0.98) = 50$ batches. The sensitivity of R@10 to each hyperparameter is reported in Appendices G.1 and G.2.

Algorithm 2: VA-NTXent: global loss with confidence-adaptive temperature

Input: Projections $\mathbf{Z}_s, \mathbf{Z}_{t1} \in \mathbb{R}^{B\times D}$ (ℓ_2-normalized)
State: Log-temperature $\log \tau_k$; confidence estimate $\hat{p}_{k-1}$
Output: Loss $\mathcal{L}^{(k)}_{\text{global}}$; updated $\log \tau_{k+1}$ and $\hat{p}_k$

$\tau_k \leftarrow \exp(\log \tau_k)$
$\mathbf{Z} \leftarrow [\mathbf{Z}_s; \mathbf{Z}_{t1}] \in \mathbb{R}^{2B\times D}, \quad s_{ij} \leftarrow \mathbf{z}_i^\top \mathbf{z}_j$ ▷ *cosine similarities*
for $i = 1, \dots, 2B$ **do**
  $i^+ \leftarrow$ index of the other view of anchor i; $\quad \mathcal{N}(i) \leftarrow \{1, \dots, 2B\} \setminus \{i, i^+\}$
  $p_i \leftarrow \exp(s_{i,i^+}/\tau_k) \,/\, \big(\exp(s_{i,i^+}/\tau_k) + \sum_{j\in\mathcal{N}(i)} \exp(s_{ij}/\tau_k)\big)$ ▷ *Eq. equation 2*
$\mathcal{L}^{(k)}_{\text{global}} \leftarrow -\frac{1}{2B} \sum_{i=1}^{2B} \log p_i$ ▷ *NT-Xent at τ_k*
$\bar{p}_k \leftarrow \frac{1}{2B} \sum_{i=1}^{2B} p_i$ ▷ *no gradient*
if $k = 0$ **then**
  $\hat{p}_k \leftarrow \bar{p}_k$
else
  $\hat{p}_k \leftarrow \lambda\, \hat{p}_{k-1} + (1-\lambda)\, \bar{p}_k$
$\log \tau_{k+1} \leftarrow \text{clip}\big(\log \tau_k + \kappa\,(\hat{p}_k - p^*),\ \log \tau_{\min},\ \log \tau_{\max}\big)$ ▷ *Eq. equation 5*
return $\mathcal{L}^{(k)}_{\text{global}}$

Algorithm 3: VA-EMA: variance-adaptive update of the local teacher

Input: Projections $\mathbf{Z}_s, \mathbf{Z}_{t2} \in \mathbb{R}^{B\times D}$ (ℓ_2-normalized); parameters $\boldsymbol{\theta}_s, \boldsymbol{\theta}_{t2}$; step k, total steps T
State: Variance estimate $\hat{v}_{k-1}$
Output: Updated teacher parameters $\boldsymbol{\theta}_{t2}$ and $\hat{v}_k$

$m_k^{\cos} \leftarrow 1 - \frac{1-m_0}{2}\big[1 + \cos(\pi k/T)\big]$ ▷ *cosine schedule*
$c_{k,i} \leftarrow \mathbf{z}_{s,i}^\top \mathbf{z}_{t2,i}$ for $i = 1, \dots, B$; $\quad v_k \leftarrow \text{Var}_i(c_{k,i})$ ▷ *no gradient*
if $k = 0$ **then**
  $\hat{v}_k \leftarrow v_k$
else
  $\hat{v}_k \leftarrow \beta\, \hat{v}_{k-1} + (1-\beta)\, v_k$
$r_k \leftarrow \text{clip}\big(v_k/(\hat{v}_k + \epsilon),\ r_{\min},\ r_{\max}\big)$ ▷ *Eq. equation 8*
$m_k \leftarrow \text{clip}\big(m_k^{\cos}(2 - r_k),\ m_{\min},\ m_{\max}\big)$ ▷ *Eq. equation 9*
$\boldsymbol{\theta}_{t2} \leftarrow m_k\, \boldsymbol{\theta}_{t2} + (1-m_k)\, \boldsymbol{\theta}_s$ ▷ *Eq. equation 7, before the Adam step*

F Evaluation on ImageNet-1K

We evaluate D-ViCe on ImageNet-1K against representative contrastive and non-contrastive self-supervised methods. All methods use a ResNet-50 backbone, are pre-trained for 200 epochs with a batch size of 256, and are evaluated by linear probing on frozen features under an identical protocol. For this experiment, we use a reduced configuration of D-ViCe that includes only the adaptive temperature and a single target encoder. As shown in Table 7, D-ViCe achieves the highest top-1 accuracy of 63.2%. It outperforms the contrastive baselines SimCLR and MoCo by 2.9 and 2.7 percentage points (pp), respectively, and the non-contrastive baselines SimSiam and BYOL by 1.2 and 1.1 pp. These consistent gains across both families, obtained using only a subset of D-ViCe's components, indicate that the proposed formulation learns more discriminative representations under a matched training budget.

Table 6: **D-ViCe hyperparameters**, identical across all CAD experiments.

Component	Symbol	Value
Global branch (confidence-adaptive temperature)		
Initial temperature	τ_0	0.1
Temperature bounds	$[\tau_{\min}, \tau_{\max}]$	$[0.05,\ 2.0]$
Reference confidence	p^*	0.65
Controller gain	κ	0.025
Confidence EMA factor	λ	0.98
Local branch (variance-adaptive momentum)		
Base momentum	m_0	0.98
Variance EMA factor	β	0.98
Variance-ratio bounds	$[r_{\min}, r_{\max}]$	$[0.5,\ 2.0]$
Momentum bounds	$[m_{\min}, m_{\max}]$	$[0.9m_0,\ 0.9999]$
Stability constant	ϵ	10^{-8}
Objective		
Local-loss weight	γ	1

Table 7: **Linear evaluation on ImageNet-1K.** Top-1 accuracy (%) of a linear classifier trained on frozen ResNet-50 features after 200 pre-training epochs (batch size 256). All methods are trained and evaluated by us under the same protocol. D-ViCe uses only the adaptive temperature and a single target encoder.

Method	Family	Top-1 (%)
SimCLR (Chen et al., 2020)	Contrastive	60.3
MoCo (He et al., 2020)	Contrastive	60.5
SimSiam (Chen & He, 2021)	Non-contrastive	62.0
BYOL (Grill et al., 2020)	Non-contrastive	62.1
D-ViCe (ours)	Contrastive	**63.2**

G Additional Ablations

All experiments in this section follow the protocol of Section 3.2: SolidLetters, R@10 with cosine retrieval on $\mathbf{h}$, one factor changed at a time, and a single run per configuration.

G.1 Sensitivity of the Temperature Controller

Table 8 varies each hyperparameter of the temperature controller (Eq. equation 5) around the default τ_0=0.1, p^*=0.65, κ=0.025, λ=0.98, $(\tau_{\min}, \tau_{\max})$=(0.05, 2.0). Across all settings, R@10 stays between 0.9443 and 0.9496, and every setting is more than 8.6 points above the baseline. The initial temperature τ_0 matters least (0.11 points across 0.05–0.3), because its effect is short-lived: in the trained model the controller holds τ_k at $\tau_{\min}$, which it reaches within the first few hundred steps from any tested τ_0. The same regime explains the other rows. Every tested reference confidence $p^* \in [0.6, 0.8]$ lies far above the positive-pair probability the model attains ($\bar{p} \approx 0.003$), so all of them drive τ_k to $\tau_{\min}$; the gain κ and the EMA factor λ only change how quickly this happens, and the upper bound $\tau_{\max}$ is never reached. Consistently, the largest range for any single hyperparameter (0.53 points) stays below the run-to-run spread (0.88 points, Appendix G.3), so we observe no effect of these choices that is distinguishable from the run-to-run spread.

Table 8: **Sensitivity of VA-NTXent.** One hyperparameter varied at a time; defaults in bold, $\tau_{\min}$=0.05 fixed. All tested p^* exceed the attainable positive-pair probability, so τ settles at $\tau_{\min}$.

Param.	value / R@10			
τ_0	0.05	**0.1**	0.2	0.3
	0.9488	**0.9496**	0.9489	0.9485
p^*	0.6	**0.65**	0.7	0.8
	0.9464	**0.9496**	0.9479	0.9466
κ	0.02	**0.025**	0.03	0.05
	0.9457	**0.9496**	0.9471	0.9445
λ	0.96	0.97	**0.98**	0.99
	0.9443	0.9454	**0.9496**	0.9475
$\tau_{\max}$	1.0	**2.0**	2.5	3.0
	0.9455	**0.9496**	0.9471	0.9484

G.2 Sensitivity of the Momentum Controller

Table 9 varies the parameters of the momentum controller (Eqs. equation 8–equation 9). A smaller m_0 lets the teacher follow the source more quickly, which weakens its role as a smoothed target, and R@10 falls to 0.9371 at m_0=0.96; a larger m_0 slows the teacher and costs 0.53 points at m_0=0.99. The factor β sets how many past batches the reference variance $\hat{v}_k$ averages over, roughly $1/(1-\beta)$: about 50 batches at the default β=0.98, but only about 17 at β=0.94. With such a short window, $\hat{v}_k$ stays close to the current batch variance, so r_k remains near 1 and the momentum rarely departs from the cosine schedule; R@10 drops by 0.92 points. At the other extreme, β=1 never updates $\hat{v}_k$ after the first batch, so every batch is compared with the variance at the start of training rather than with the preceding batches; as the representation changes during training, r_k then drifts away from 1 for reasons unrelated to the current alignment, and R@10 drops by 0.67 points. The observed results favor comparing each batch with an average over the last few dozen batches. Besides γ, only m_0 (1.25 points) and, marginally, β (0.92 points) change R@10 by more than the run-to-run spread.

Table 9: **Sensitivity of VA-EMA.** One hyperparameter varied at a time; defaults in bold, $r_{\max}$=2 fixed. β averages $\hat{v}_k$ over $\approx 1/(1-\beta)$ batches; β=1 keeps the first batch's value.

Param.	value / R@10			
m_0	0.96 0.9371	0.97 0.9424	**0.98** **0.9496**	0.99 0.9443
β	0.94 0.9404	0.96 0.9436	**0.98** **0.9496**	1.0 0.9429
$r_{\min}$	0.3 0.9450	**0.5** **0.9496**	0.6 0.9473	0.7 0.9408

G.3 Run-to-Run Variation

The $r_{\min}$ row of Table 9 provides an empirical estimate of run-to-run spread. With $m_0 \geq 0.98$, any $r_k \leq 2 - m_{\max}/m_0 \approx 0.98$ already drives m_k to $m_{\max}$ (Remark 1(c)). Every tested $r_{\min} \in \{0.3, 0.5, 0.6, 0.7\}$ lies below this threshold, so all four settings produce identical momentum at every step and differ only by run-to-run randomness. Their range, 0.9408–0.9496 (0.88 points), is therefore an empirical run-to-run spread over four runs, not a formal variance estimate. Differences smaller than this, which include most entries in Tables 8 and 9, are not distinguishable from this spread and should not be read as evidence that one value is optimal.

G.4 Batch Size

Table 10 varies the batch size B, which also sets the number of negatives seen by the global objective. R@10 rises by 0.75 points from B=128 to B=256 and then stays within 0.05 points up to B=1024, well inside the run-to-run spread. Beyond 256 graphs, additional negatives therefore bring no measurable benefit, and we use B=256 as the smallest batch size that reaches this level.

Table 10: **Batch size.** Each anchor sees $2B-2$ negatives.

B	**Negatives**	**R@10**
128	254	0.9421
256	510	**0.9496**
512	1022	0.9494
1024	2046	0.9491

H Retrieval Accuracy at Scale

To complement the recall metrics of Table 1, we count how many of the ten retrieved models are correct for every query. We use all 9,728 test solids as queries against a gallery of 31,285 training embeddings, retrieve the ten nearest models per query, which gives 97,280 retrievals per method, and average the number of correct retrievals over the retrieval back-ends of Section 3.1.

Fig. 5 shows that D-ViCe obtains 71,076 correct retrievals, a precision@10 of 73.1%. The strongest baselines in this measure are MoCo (He et al., 2020) with 40,638 (41.8%) and SimCLR (Chen et al., 2020), the objective of the original UV-Net, with 37,918 (39.0%), so D-ViCe retrieves 74.9% more correct models than MoCo and 87.4% more than SimCLR. The remaining methods follow: VICReg (Bardes et al., 2021) (34,138; 35.1%), SimSiam (Chen & He, 2021) (34,137; 35.1%), SwAV (Caron et al., 2020) (32,281; 33.2%), DINO (Caron et al., 2021) (27,083; 27.8%), Barlow Twins (Zbontar et al., 2021) (25,227; 25.9%) and BYOL (Grill et al., 2020) (23,674; 24.3%).

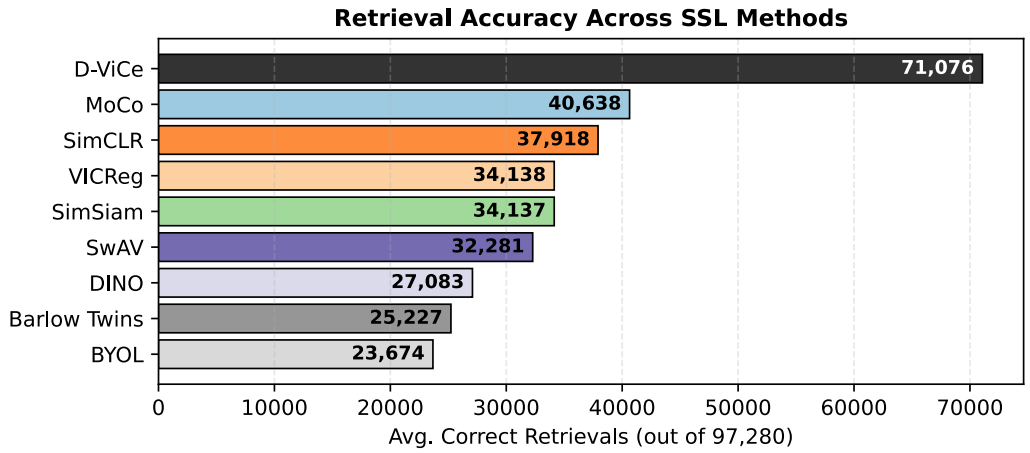

Figure 5: **Correct top-10 retrievals** out of 97,280 (9,728 test queries $\times$ 10), averaged over the retrieval back-ends of Section 3.1. †BYOL without its predictor head, as in Table 1.

The gap is therefore larger here than in the recall metrics: R@10 only asks whether one correct model appears among the top ten, whereas precision@10 counts every correct model in the top ten. D-ViCe places on average about seven correct models among the ten nearest neighbors, while every baseline places at most about four, indicating that its embedding groups same-class solids tightly rather than merely placing one of them near the query. The ranking of the baselines differs from Table 1 (MoCo is second here, SimCLR is second in R@10), which again shows that single retrieval metrics give an incomplete picture.

I t-SNE Projection Comparison

Fig. 6 visualizes the embeddings of all nine methods with t-SNE. As a two-dimensional projection, t-SNE does not preserve global distances, so we use it only as a qualitative complement to the retrieval metrics. The D-ViCe embedding (top left) forms compact, well-separated class clusters, consistent with its high precision@10 (Appendix H). Among the baselines, MoCo (He et al., 2020) and SimCLR (Chen et al., 2020) (top row) show less coherent organization, with several class groups overlapping. BYOL (Grill et al., 2020), SimSiam (Chen & He, 2021) and SwAV (Caron et al., 2020) (middle row) produce more diffuse or elongated patterns, and Barlow Twins (Zbontar et al., 2021) and VICReg (Bardes et al., 2021) (bottom row) produce broad, manifold-like structures that do not translate into tight class-level grouping. DINO (Caron et al., 2021) (bottom right) captures a recognizable coarse structure but still mixes several classes. Several methods thus recover meaningful structure, but D-ViCe gives the clearest separation, in agreement with the quantitative results.

J Qualitative Comparison with the UV-Net Baseline

Fig. 7 compares the ten nearest retrievals of D-ViCe and of the UV-Net SimCLR baseline (Jayaraman et al., 2021) for six queries. The queries were selected at random from the test set. For all six queries, the ten models retrieved by D-ViCe belong to the correct class and preserve the key geometric structure of the query, including stroke layout and overall proportions. The SimCLR baseline returns several incorrect models (red boxes), typically shapes that resemble the query locally but differ in global structure or topology, such as a different number or arrangement of strokes. This matches the quantitative gap in precision@10 (Appendix H): D-ViCe separates classes by global structure, whereas the baseline is more easily misled by local similarity. Interactive 3D views of these queries and their retrievals are provided in the `visualization` folder of our anonymous code repository.

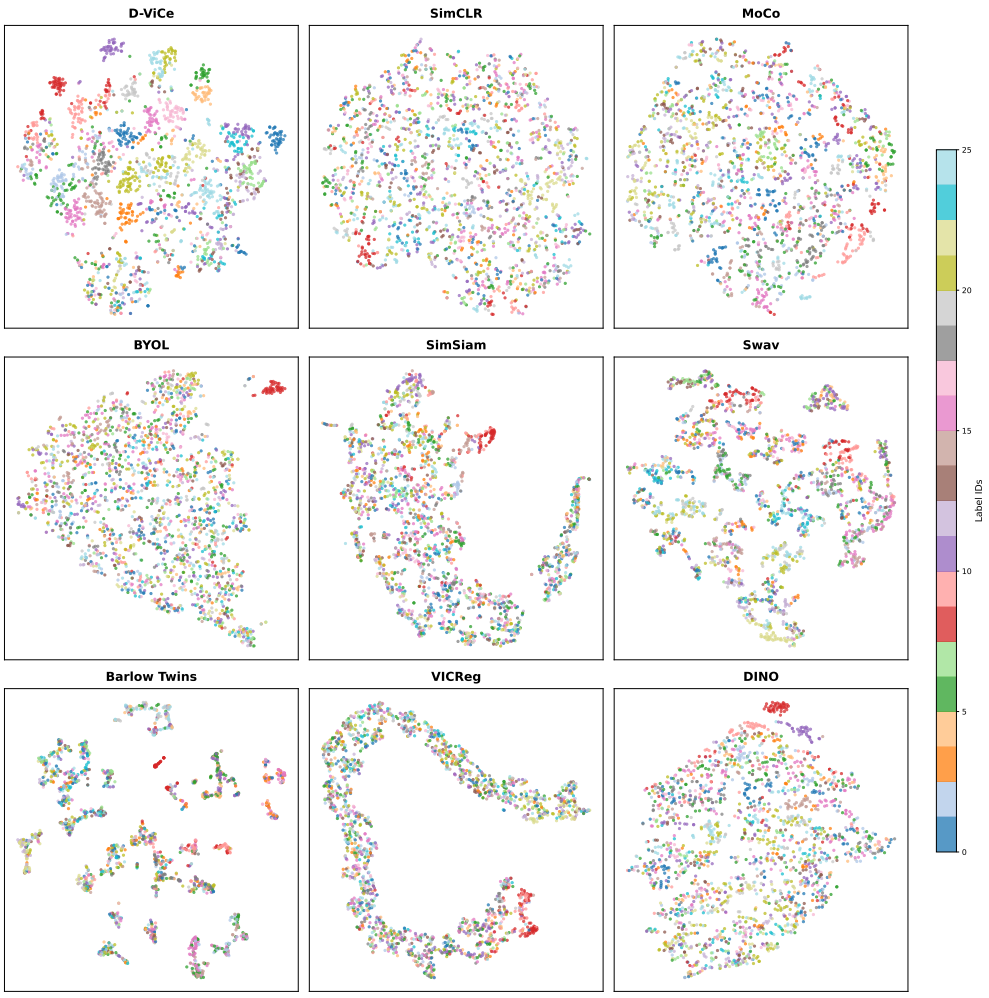

Figure 6: **t-SNE projections** of the embedding $\mathbf{h}$ learned by D-ViCe and eight self-supervised baselines on SolidLetters test solids; colors denote letter classes.

K Consistency across Retrieval Back-Ends

Fig. 8 shows the 21 nearest retrievals of D-ViCe for four queries, two upper-case (A, F) and two lower-case (r, f) letters, under four retrieval implementations. Two of them compute exact cosine similarity (a PyTorch top-k and a scikit-learn implementation), one performs exact Euclidean search with a flat FAISS index, and one performs approximate cosine search with an HNSW graph index, so the figure compares two distance measures and exact against approximate search. For each query, the retrieved sets match the query's letter class and preserve its edge layout, volumetric proportions and overall spatial configuration, including for queries with subtle shape variations. All four implementations return nearly identical sets, differing only in the order of near-tied neighbors. The agreement between cosine and Euclidean search and between exact and approximate search indicates that the neighborhood structure is a property of the learned embedding rather than of the distance measure or the index.

L Design Rationale: Questions and Answers

This section collects the questions we expect readers to ask about D-ViCe's design, together with the evidence behind each answer. Table 11 summarizes the main choices.

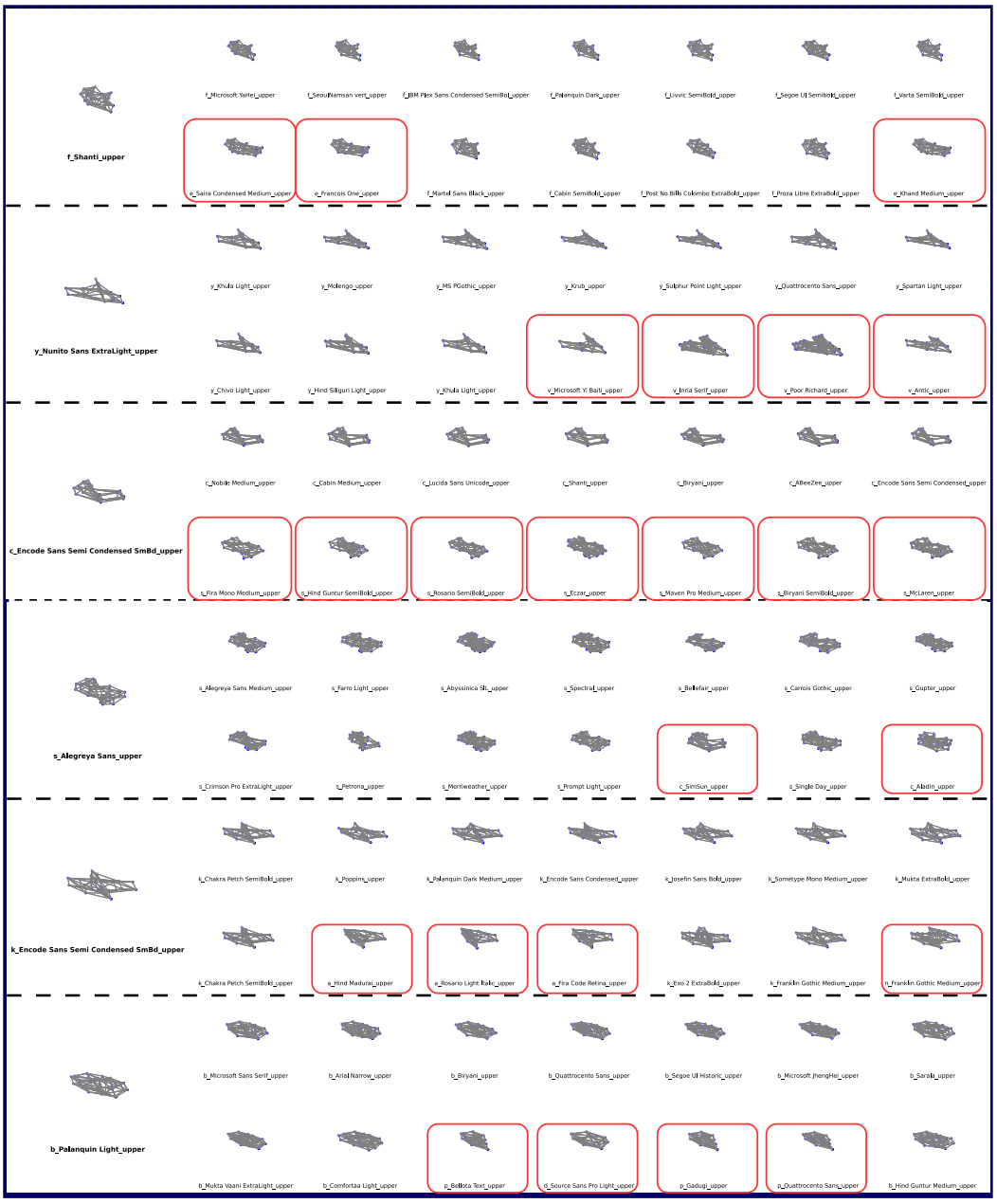

Figure 7: **Top-10 retrievals for six queries** with D-ViCe (upper row of each pair) and the UV-Net SimCLR baseline (Jayaraman et al., 2021) (lower row). Red boxes mark incorrect retrievals.



Figure 8: **Top-21 retrievals of D-ViCe with four retrieval implementations through our developed GUI** for the queries A, F, r and f: exact cosine top-k (PyTorch and scikit-learn), exact Euclidean search with FAISS (`IndexFlatL2`), and approximate cosine search with HNSW (M=16, $ef_{\text{construction}}$=100, ef=20).

Table 11: **Design decisions of D-ViCe**, the alternative we compared against, and where the evidence is reported.

Choice in D-ViCe	Alternative	Evidence
Frozen random global teacher	EMA or variance-adaptive EMA teacher	Table 4
NT-Xent for the global branch	Fixed-temperature NT-Xent without the frozen teacher	Table 3, rows 1–2
Local branch with EMA teacher	No local branch ($\gamma = 0$)	Tables 3 and 5
Variance-adaptive momentum	Cosine schedule only	Table 3
Equal weighting, $\gamma = 1$	$\gamma \in [0, 1.4]$	Table 5
Confidence-adaptive temperature	Controller hyperparameter sweeps	Table 8, Sec. 2.2

Q1. Why NT-Xent for the global branch but a cosine-similarity loss for the local branch? The two branches do different jobs. NT-Xent introduces inter-instance discrimination through negatives, including source–source pairs that penalize a constant embedding, whereas the local cosine-similarity objective provides only positive-pair alignment with a smoothly updated teacher. Table 5 shows that the combination matters: removing the local term ($\gamma = 0$) costs 2.52 points, and weighting it more heavily than the global term ($\gamma = 1.4$) empirically reduces R@10 by 0.98 points.

Q2. Why do both branches use the same augmentations? Using the same views for both teachers means that the two branches differ only in their target and their loss, so the ablations in Section 3.2 isolate exactly these factors. It also keeps D-ViCe a drop-in replacement for the UV-Net objective, since we reuse UV-Net's augmentations unchanged. We did not explore branch-specific augmentations; whether a stronger view for one branch helps is an open question that we list among our limitations (Appendix M).

Q3. Why is only the local teacher updated, while the global teacher stays frozen? Because the two branches need opposite kinds of targets. For instance discrimination, a stationary target is the better reference: letting the global teacher follow the source by EMA lowers R@10 by 4.76 points, and a variance-adaptive update lowers it by 6.07 points (Table 4). For alignment, a target that tracks the source is needed, and there the same variance-adaptive update helps (Table 3). Keeping one teacher frozen and one adaptive is therefore a deliberate split of roles, not an asymmetry for its own sake.

Q4. Is a randomly initialized teacher not uninformative? It does not need to be informative on its own. The frozen teacher provides targets that depend on the input and never change during training, while NT-Xent supplies the instance-discrimination signal: the source must make the other view of each solid identifiable among $2B - 1$ candidates. Earlier work reports that predicting or distilling a random network can yield useful representations (Grill et al., 2020; Sarnthein et al., 2023), and Table 4 shows that on B-rep graphs this frozen target outperforms EMA-updated alternatives.

Q5. Does the temperature controller actually adapt the temperature? Only briefly. On B-rep graphs the positive-pair probability stays far below the reference value, so the controller lowers the temperature to $\tau_{\min}$ within the first few dozen steps and keeps it there (Section 2.2); Proposition 1 explains why no interior operating point exists in this regime. In our experiments the controller therefore acts as a short, data-driven decay to a fixed low temperature, and its hyperparameters have little effect (Table 8). We report this openly and discuss a chance-relative reference that would keep it adaptive in Appendix M.

Q6. Is the improvement from VA-EMA significant? Not on its own. VA-EMA improves R@10 by 0.70 points over a cosine-scheduled EMA teacher, which lies within the run-to-run spread of about 0.9 points (Appendix G.3). The large gains of D-ViCe come from the frozen global teacher and from adding the local branch, both of which exceed this spread several times over.

Q7. What does D-ViCe cost compared with SimCLR? Only training becomes more expensive. D-ViCe keeps two target networks in memory, and each adds one forward pass per step but no backward pass. Only the source encoder is deployed, so the model used for retrieval has the same size and speed as the UV-Net baseline.

Q8. How sensitive is D-ViCe to its controller hyperparameters? On SolidLetters, most controller hyperparameters change R@10 by less than the observed run-to-run spread (Appendices G.1 and G.2); besides γ, the base momentum m_0 has the largest effect. We used the same values for every CAD experiment (Table 6), but have not tested their transfer to other CAD corpora (Appendix M).

M LIMITATIONS

Reliability of the variance signal. VA-EMA uses the batch variance of source–teacher similarities, $v_k = \mathrm{Var}_i(c_{k,i})$, as a proxy for the batch-wise heterogeneity of source–teacher alignment; it does not measure the overall alignment quality, since a uniformly poor alignment can have low variance. This proxy can become noisy when the batch is small, when augmentations are highly stochastic, or when a batch contains many outliers. In these cases v_k fluctuates sharply relative to its running estimate $\hat{v}_k$, and the controller may overreact: the momentum oscillates or stays at its clipping bounds, which yields an unstable target and can degrade the consistency of the local branch. The clipping of r_k and m_k (Eqs. equation 8–equation 9) bounds this effect but does not remove it. Even in regular training the controller is strongly asymmetric: for the trained model, about two thirds of the batches have $m_k = m_{\max}$ (Remark 1(c)), so the teacher advances mainly in bursts.

Saturation of the temperature controller. The temperature feedback is corrective only when the reference confidence p^* is attainable (Proposition 1). On B-rep graphs it is not: the trained model attains a positive-pair probability of only $\bar{p} \approx 0.003$, so the controller reaches $\tau_{\min}$ within the first few dozen steps and stays there (Section 2.2). In our experiments it therefore acts as a short, data-driven decay to a fixed low temperature rather than as a continuously adapting one. A reference defined relative to the chance level $1/(2B-1)$ would keep the controller in its corrective regime; we leave this to future work.

Size of individual gains. Each ablation configuration is a single training run, and the run-to-run spread is about 0.9 points of R@10 (Appendix G.3). The largest effects, from the global branch ($+6.66$ points) and from the frozen teacher (4.76–6.07 points), are well above this level, whereas the gain of VA-EMA over a standard EMA teacher ($+0.70$ points) lies within it. Likewise, the improvements on the TU benchmarks are consistent in direction but modest, exceeding one standard deviation in 16 of 54 method–dataset pairs.

Scope of the evaluation. Our quantitative CAD results use SolidLetters; ABC is evaluated qualitatively. Whether the gains carry over to other CAD corpora, to other B-rep encoders, and to other modalities remains to be tested. We also keep the augmentations of UV-Net fixed and do not study how D-ViCe interacts with different augmentation choices.

Training cost. D-ViCe keeps two target networks in memory alongside the source encoder, and each adds one forward pass per training step. Inference is unaffected, since only E_s is deployed, but training memory and time grow compared with single-network objectives such as SimCLR.