Geometric transfer · 3D point clouds

Amanuel Tsehay · amanuelkks@gmail.com

1 · PROBLEM

what and why

Does large-scale self-supervised pretraining on a generic 3D shape corpus (ShapeNet) learn a transferable geometric prior that measurably narrows the well-documented synthetic-to-real performance gap in point cloud recognition — and if so, by how much, and where does the benefit concentrate? The synthetic-to-real gap in 3D point cloud learning is a known, quantified problem: models trained purely on clean synthetic CAD data (ModelNet40) suffer severe accuracy drops when evaluated on real- world scanned objects (ScanObjectNN), which carry sensor noise, occlusion, and background clutter that synthetic data lacks. Closing this gap through better representation learning — rather than more labeled real data, which is expensive to collect — remains an open problem. This project asks whether a self-supervised geometric pretraining objective, learned at scale on unlabeled shapes, is one viable lever. Point cloud models trained and evaluated in-domain on synthetic CAD data now perform very well. The same models degrade sharply when the input distribution shifts to real-world scans — noisy, partial, cluttered with background points. Most existing work closes this gap with architectural tricks or heavier supervision on real data. Comparatively less work asks whether a large-scale self-supervised pretraining learned once on generic shapes, transfers a general notion of 3D surface structure that helps regardless of downstream domain.

2 · APPROACH

model / method, and why this one

Stage 1 — Self-supervised pretraining on ShapeNet. Train a permutation-invariant point cloud encoder (PointNet++-style set abstraction backbone) using two geometric pretext tasks: Masked point cloud completion: randomly occlude a contiguous region of each shape’s point set, train the encoder-decoder to reconstruct the missing region, scored with Chamfer distance between predicted and ground-truth point positions. Rotation-invariant contrastive learning: present the encoder with two randomly rotated views of the same shape and a different shape, train it to embed same-shape views closer than different-shape views, independent of orientation. Stage 2 — Downstream evaluation. Freeze (then optionally fine-tune) the pretrained encoder and attach a lightweight classification head. Evaluate three conditions: 1. Trained from scratch on ModelNet40, tested on ModelNet40 (in-domain baseline) 2. Trained from scratch on ModelNet40, tested on ScanObjectNN (the known sim-to-real gap, no pretraining) 3. ShapeNet-pretrained encoder, fine-tuned on ModelNet40, tested on ScanObjectNN (the actual research question) The gap between condition 2 and condition 3 is the core result.

3 · DATASET

source, size, preprocessing

Dataset Role Size ShapeNetCore Self-supervised ~51,300 3D CAD models, 55 categories pretraining ModelNet40 Downstream training 12,311 models, 40 categories (9,843 train / 2,468 test) / in-domain eval ScanObjectNN Real-world out-of- 2,902 real scanned objects, 15 categories (~2,309 train / 581 test), multiple domain eval perturbation variants (OBJ_ONLY, OBJ_BG, PB_T50_RS) All three are public, standard benchmarks with no data-use agreement or institutional gatekeeping — available immediately, no access delay on the critical path.

4 · METRICS

how you’ll know it worked

Overall accuracy (OA) and mean per-class accuracy on ModelNet40 (in-domain sanity check) OA on ScanObjectNN, across its three difficulty variants (object-only, with background, hardest perturbed) Sim-to-real gap: the accuracy delta between in-domain and out-of-domain performance, with and without pretraining — this delta, not the raw accuracy, is the primary reported result Comparison against published from-scratch baselines on the same benchmarks for context

5 · COMPUTE

`training time, memory, dependencies`

Encoder size: ~1.5–3.5M parameters (PointNet++-scale) — modest by modern standards Pretraining: full ShapeNet pass, multiple epochs, single-GPU multi-day job (or shortened on multi- GPU) Fine-tuning/eval sweeps across three conditions and multiple ScanObjectNN variants Estimated VRAM: 6–10 GB per training run — comfortably within a single AMD accelerator, using a small fraction of available memory. This is a single-GPU-scale project; it does not require multi-GPU allocation. Primary compute need is sustained single-GPU time for the pretraining run, not peak memory — a request that’s easy to schedule around larger jobs in the group.

6 · DELIVERABLES

`model, paper, demo`

A reusable self-supervised point cloud pretraining pipeline (code, released openly) A trained ShapeNet-pretrained encoder checkpoint Benchmarked results across all three evaluation conditions, with sim-to-real gap quantified A short paper suitable for a NeurIPS-adjacent workshop or similar venue, with AMD acknowledged for compute support

7 · RISK

`what breaks first, and plan B`

what breaks first, and plan B benchmarks is already well-documented in prior work, a null or small result is still a legitimate, citable finding — it quantifies how much (or how little) one specific intervention moves a known, measured problem. This is not a dead end; there is no scenario where the core experiment produces an unpublishable result. ability to report results from partial pretraining if needed. landmark/graph-based encoder (GNN over a sparser point set) if the full PointNet++-style pipeline proves unstable to train — a legitimate, still-geometric alternative. STRETCH GOAL — CONDITIONAL, NOT PART OF CORE DELIVERABLES If Stage 1 pretraining shows a meaningful transfer benefit on the core benchmarks, the same pipeline — encoder architecture and training loop already built and validated — could be pointed at a small public 3D face landmark dataset (e.g. Bosphorus or BU-3DFE) to test whether the same generic geometric prior extends to a much more constrained, fine-grained shape category. This is not counted in the compute request or core timeline; it is a natural low-cost extension using infrastructure the core project already produces. References Qi et al. — PointNet and PointNet++, foundational point cloud deep learning architectures Wu et al. — ModelNet, the standard synthetic CAD benchmark Uy et al. — ScanObjectNN, the standard real-world point cloud benchmark and its documented domain gap from synthetic training Chang et al. — ShapeNet, the large-scale annotated 3D model repository used for pretraining