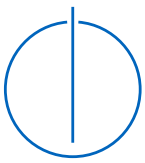
SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Mining of Reusable Component Libraries through Unsupervised Multi-Modal Clustering

Mustafa Sercan Amac



SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY

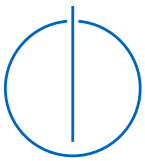
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Mining of Reusable Component Libraries through Unsupervised Multi-Modal Clustering

Erschließung wiederverwendbarer Bauteilbibliotheken durch unüberwachtes multimodales Clustering

Author:	Mustafa Sercan Amac
Examiner:	Prof. Dr.-Ing. habil. Alois C. Knoll
Supervisor:	Prof. Dr.-Ing. André Borrmann & Panagiotis Petropoulakis
Advisor:	Georgios Pavlidis
Submission Date:	08.06.2026



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 08.06.2026 Mustafa Sercan Amac

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Prof. Dr.-Ing. André Borrmann (Chair of Computing in Civil and Building Engineering, TUM School of Engineering and Design) and Prof. Dr.-Ing. habil. Alois C. Knoll (Robotics, Artificial Intelligence and Real-time Systems, TUM School of CIT), for the opportunity to pursue this thesis and for the consistent academic guidance throughout. Special thanks go to my day-to-day advisor, Panagiotis Petropoulakis (TUM Georg Nemetschek Institute / Leonhard Obermeyer Center), whose patient feedback, sharp critiques of methodology, and willingness to revisit assumptions repeatedly shaped the work into its present form.

I am grateful to Georgios Pavlidis at Bentley Systems, our industry partner, whose practical perspective on real BIM authoring workflows kept the research grounded in genuine library-mining needs rather than benchmark-chasing. The collaboration with Bentley enabled the framing of the thesis title around *reusable component libraries* as a tangible deliverable.

The TUM Georg Nemetschek Institute and the Leonhard Obermeyer Center provided not only the computational and data resources but also the intellectual environment in which a BIM + multi-modal-ML project becomes possible.

Finally, I would like to thank my family for their unwavering support throughout this journey.

Abstract

Federated Building Information Modeling (BIM) repositories typically contain tens to hundreds of thousands of objects whose authored Industry Foundation Classes (IFC) type labels are noisy, inconsistent across submissions, and frequently degraded to schema catch-alls such as `IfcBuildingElementProxy`. As a consequence, the reusable component library that a project actually contains is buried inside the data: designers cannot find existing parts, search is keyword-based, and library-curation tooling depends on the modeller's intent at authoring time rather than on what an object physically is.

This thesis introduces an end-to-end, reproducible pipeline that *mines* this latent library from raw IFC submissions without supervision. We compose three modalities per element: nine handcrafted geometric features built from a four-stage engineering progression (orientation, cross-section aspect ratio, normal entropy, horizontal-face fraction); free-form text descriptions produced by Gemini 3.1 Flash Lite under a seven-version prompt-engineering study, from generic geometric tags (v1) to IFC-aware descriptions (v7), embedded with Gemini Embedding 2 (768-d); and image embeddings from three modern visual foundation models (DINOv3, SigLIP2, DuoDuo CLIP) computed on nine canonical-view renders per object. A family of fusion recipes, spanning per-block PCA and UMAP, joint PCA and joint UMAP over the raw and the variance-balanced concatenation, late Borda rank fusion and canonical correlation, is ablated against macro purity and recall@10.

On a curated benchmark of **1,700 manually annotated objects across 17 merged IFC classes**, extracted from 202 TUM student IFC submissions, the winning F5_varbal recipe (variance-balanced concatenation of the three modality blocks, projected to four dimensions with UMAP) reaches **macro purity** 0.853 **at** $k = 126$ **clusters** (bootstrap mean 0.848 ± 0.008) with HDBSCAN (leaf selection, mcs=5, ms=3, force-assignment), and 0.863 ± 0.006 with K-Means at the same k discovered by HDBSCAN: the role of HDBSCAN here is automatic granularity discovery, not partition quality. The bootstrap adjusted-Rand-index is 0.874 ± 0.019 on $10\times$ 90% subsamples. The recovered partition is more than seven times finer than the 17-class schema grain, surfacing component sub-types that the IFC type label alone cannot represent. Late Borda rank fusion over the same three modalities reaches **recall@10 = 0.966** for similarity retrieval. The complete codebase, persisted result tables and figures reproduce in a few hours on a

single laptop.

Contents

1 Introduction

1.1 Background and Motivation

Building Information Modeling (BIM) has, over the past two decades, become the de-facto digital backbone of the Architecture, Engineering and Construction (AEC) industry. A modern BIM authoring workflow does not stop at documenting geometry: it instantiates every wall, slab, beam, column, door, window, fixture or duct as a typed object carrying a hierarchy of properties, relations, materials and quantities. The open standard that formalises these objects, the Industry Foundation Classes (IFC) schema maintained by buildingSMART International, defines several hundred entities and an even larger number of pre-defined type and predefined-type enumerations. In principle, this gives the AEC practitioner a complete machine-readable description of the asset: *what* each element is, *where* it sits, *how* it relates to its neighbours, and *with what* it is built.

In practice the picture is far less tidy. Real federated BIM models routinely contain tens to hundreds of thousands of objects authored across multiple disciplines, by multiple firms, in multiple authoring tools, over multiple years. Two phenomena make this scale unmanageable for downstream use:

- **Reusable components are buried.** A 50-storey project will typically contain only a few dozen *distinct* structural cross sections, a handful of door families and a small library of M.E.P. fittings, but each appears hundreds or thousands of times in slightly altered placements. Designers who want to find "the standard precast spandrel we used on level 4" must rely on filename conventions, library tags, or keyword search, all of which depend on the modeller's intent at authoring time rather than on what the object actually *is*.
- **IFC types are unreliable as semantic anchors.** The same physical artefact can legitimately be labelled `IfcMember`, `IfcBeam`, `IfcBuildingElementProxy`, or even `IfcBuildingElementPart` depending on which authoring tool was used, how the family was instantiated, and which mapping table was selected on export. `IfcBuildingElementProxy` alone is so generic that it functions as a catch-all bucket for "anything the exporter could not classify". Type labels are therefore a

noisy, partial signal: useful for navigation, but unreliable as the basis for retrieval, quality control or library curation.

The consequence is a paradox: BIM repositories are simultaneously *data-rich* (every face, every property, every relation is recorded) and *insight-poor* (the structure that would let an architect actually reuse this data is locked behind inconsistent labels). What is needed is a mechanism that can look at a BIM repository, ignore the modeller's choice of words, and surface the latent library of distinct reusable components that the project actually contains.

1.2 Problem Definition

This thesis investigates the following problem:

> *Given a large heterogeneous collection of 3D building element meshes extracted from federated IFC models, with noisy or missing type labels, can we recover, without supervision, a reusable component library that (i) reflects geometric, visual and semantic identity, (ii) generalises across authoring conventions and (iii) supports downstream library operations such as catalog generation and similarity retrieval?*

We frame this as an *unsupervised multi-modal clustering* task. Three modalities, namely handcrafted geometric features, free-form text descriptions produced by a vision–language model, and image embeddings from large pretrained visual encoders, are computed for every object, projected into a shared low-dimensional space, and clustered.

The dataset is a curated subset of **128,883 raw meshes** extracted from **202 student IFC submissions** of a TUM Computational Modeling & Simulation course (Prof. Borrmann's group). After per-type representative sub-sampling and manual annotation we obtain **1,700 objects across 17 merged IFC classes** (see Section 4.2 for the full pipeline). This is the benchmark on which every clustering configuration in this thesis is evaluated.

1.3 Research Questions

The investigation is organised around five concrete research questions:

RQ1 *How far can purely geometric handcrafted features take us?* Starting from a 4-feature Oriented Bounding Box (OBB) baseline (length, cross-section, volume, vertex count) we ask which targeted engineered features (orientation, cross-section aspect ratio, normal entropy, horizontal fraction) recover which IFC-type confusions, and to what extent geometry alone can mine the library.

RQ2 *Can language describe BIM objects well enough to cluster them?* We render every object from nine canonical viewpoints, pass the multi-view image set through a vision–language model (Gemini 3.1 Flash Lite), receive a structured shape description in JSON, embed the text with Gemini Embedding 2 (768-d), and cluster.

RQ3 *Do general-purpose visual encoders provide a competitive signal?* We extract image embeddings from three modern encoders, SigLIP2 (1024-d), DINOv3 (1024-d) and DuoDuo CLIP (512-d), on both coloured and colourless renders, and compare them to the geometric and textual baselines.

RQ4 *What is the right way to fuse these modalities?* We design and ablate seven fusion recipes spanning per-block PCA and per-block UMAP, joint PCA and joint UMAP over the raw and the variance-balanced concatenation, and late Borda rank fusion. We seek the fusion that maximises macro purity (clustering) *and* recall@10 (retrieval) on a fixed evaluation set.

RQ5 *How far is the unsupervised winner from a supervised ceiling, and how stable is the resulting partition?* We train a Random Forest with 5-fold cross-validation on the same features to bound the achievable macro recall, then run 10× bootstrap subsampling to quantify Adjusted-Rand-Index stability of the unsupervised pipeline.

1.4 Contributions

This thesis makes the following concrete contributions:

1. **A reproducible end-to-end pipeline** (Python, `scikit-learn`, `trimesh`, `ifcopenshell`, `umap-learn`, `hdbscan`) that ingests raw IFC submissions, extracts meshes, computes nine handcrafted geometric features, renders nine canonical views per object, generates Gemini-VLM descriptions, embeds them, extracts visual features from three modern encoders, fuses the three modalities and clusters the result. After a one-time visual-encoder extraction (1.2–3.5 h per encoder on a Colab T4 GPU, cached to disk), downstream re-runs complete in a few hours on a single Apple-Silicon laptop.
2. **A curated benchmark** of 1,700 manually annotated objects across 17 merged IFC classes, derived from 202 real student submissions with their idiosyncratic modelling conventions. The annotation pool was selected via per-type K-Means representative sub-sampling, a one-shot scheme that snaps each per-type cluster centroid to its nearest real mesh and empirically matches full-dataset clustering metrics within noise.

3. **A four-stage geometric feature-engineering progression** (orientation, then cross-section aspect ratio, then normal entropy, then horizontal fraction) lifting macro purity from 0.434 (4 features) to 0.704 (9 features) at k=32, with each stage targeting an explicit IFC-type confusion identified at the previous stage. Every feature is justified by analysis-of-variance F statistics, per-type purity delta and confusion-heatmap evidence.

4. **A seven-version prompt-engineering study** (v1 generic geometric tags through v7 IFC-aware) showing that text-only clustering rises from macro 0.534 (v1) to 0.740 (v7) when the prompt is steered to surface schema-aware terminology, with UMAP-8 universally beating PCA on textual embeddings.

5. **A 30-combination HDBSCAN sweep, a 10-reducer ablation per modality, and a seven-recipe fusion comparison** (F1a, F1a_umap, F1b, F1b_varbal, F5, F5_varbal, CCA) totalling ≈29,400 clustering runs. The winning recipe is F5_varbal: the variance-balanced concatenation of the three modality blocks projected to four dimensions with UMAP, evaluated by HDBSCAN at leaf cluster selection, `mcs`=5, `ms`=3 with force-assignment.

6. **A k-matched comparison** between K-Means and HDBSCAN on the fusion winner, settling the question of whether HDBSCAN's apparent advantage is a partition-quality or a granularity-selection effect. At the $k = 126$ that HDBSCAN discovers from the data, K-Means reaches macro purity 0.863 ± 0.006 and HDBSCAN-force 0.853: the headline number is therefore a property of the F5_varbal representation at the data-natural granularity, with HDBSCAN contributing *automatic granularity discovery* (k=17 $\rightarrow$ 126 is +25 pp of macro purity) and HDBSCAN-force additionally guaranteeing a complete no-noise partition for the catalog deployment.

7. **Two downstream library applications**: (a) an auto-generated catalog where each of the 126 clusters becomes a navigable entry with three exemplars, a full member gallery and TF-IDF keywords; and (b) a late-Borda retrieval pipeline achieving **recall@10 = 0.966** across the 1,700 objects.

8. **Stability and ceiling analysis**: bootstrap Adjusted Rand Index (ARI) of 0.874 ± 0.019 on the headline winner and a Random Forest (RF) supervised ceiling of 0.857 macro recall on the winning representation, an indicative upper bound on the label-consistent structure it contains.

9. **A cross-corpus validation** on the public IFCNetCore benchmark (Chapter 6), re-running every modality and fusion recipe on 7,930 objects across 20 native IFC

classes to confirm that the engineered features and the recipe ranking are not over-fitted to the TUM corpus.

1.5 Scope and Limitations

This thesis deliberately bounds its scope in three ways:

- **Two corpora.** The headline results are established on the TUM student-submission corpus; Chapter 6 re-runs the pipeline on the public IFCNetCore benchmark as an external-validity check. We do not yet evaluate on industrial-scale federated repositories. Section 7.2 returns to this.
- **17 merged classes.** The original 24 IFC types contain pairs of near-synonyms (e.g. `IfcWall` vs. `IfcWallStandardCase`) which we merge for evaluation purposes via a `TYPE_MAP`. Two schema catch-all types (`IfcBuildingElementProxy`, `IfcBuildingElementPart`) are excluded because they carry no geometric identity. The merged grain is what the schema can actually distinguish, not the underlying number of *distinct* reusable components, which the mining pipeline subsequently discovers to be on the order of 100 (see Section 5.6).
- **Unsupervised throughout.** We never train a network on labels for the main result. The RF supervised ceiling is used *diagnostically* to bound the gap, not as a deployment target.

1.6 Thesis Outline

The remainder of this document is organised as follows:

Chapter 2, *Theoretical Background* , introduces the foundations: BIM and IFC, 3D mesh representation, OBB versus Axis-Aligned Bounding Box (AABB), principal component analysis, the clustering algorithms we use (K-Means, Bisecting K-Means, GMM, HDBSCAN), dimensionality reduction (PCA, UMAP), and the modern visual and textual embedding models (DINOv3, SigLIP2, DuoDuo, Gemini).

Chapter 3, *Related Work* , surveys prior art across four threads: 3D shape descriptors for BIM, IFCNet-style supervised classification, language-grounded 3D understanding, and recent MCP/agentic frameworks for IFC.

Chapter 4, *Methodology* , is the main technical chapter: it walks through dataset extraction and annotation, the geometric feature-engineering progression with full mathematics, the 9-viewpoint render pipeline, the prompt engineering for VLM descriptions (v1 through v7), the three visual encoders, the seven fusion recipes, the clustering search space, and the evaluation protocol.

Chapter 5, *Results and Discussion* , reports every experimental result with figures and tables: the 4-feature baseline, the four-stage feature engineering gains, the algorithm comparison, the text-prompt progression, the visual-encoder bake-off, the fusion ablation, the k-matched comparison, the supervised ceiling, the stability analysis, and the downstream applications.

Chapter 6, *Cross-Corpus Validation on IFCNetCore* , re-runs every modality (geometry, vision, text), every fusion recipe and the supervised ceiling on the public IFCNetCore benchmark (7,930 objects across 20 native IFC classes) and confirms that the fusion-recipe ranking generalises across datasets (Spearman ≈ 0.94).

Chapter 7, *Conclusion and Future Work* , summarises the findings against each research question, lists the immediate next steps (industrial-scale validation, agentic retrieval, deep-learning frontiers) and the longer-term outlook.

Chapter 8, *Supplementary Material* , collects the full sweep tables, the reproducibility checklist, and the raw per-type purity tables.

1.7 Collaboration Context

This thesis is jointly supervised by Prof. Dr.-Ing. André Borrmann (Chair of Computing in Civil and Building Engineering, TUM School of Engineering and Design, supervising chair) and Prof. Dr.-Ing. habil. Alois C. Knoll (Robotics, Artificial Intelligence and Real-time Systems, TUM School of CIT, examiner). The advisor is Panagiotis Petropoulakis (TUM Georg Nemetschek Institute / Leonhard Obermeyer Center). The work was carried out in collaboration with Bentley Systems (industry partner; Georgios Pavlidis). The midterm presentation was delivered in Munich on 15 May 2026.

2 Theoretical Background

This chapter introduces the conceptual machinery that the rest of the thesis will compose: the data domain (BIM, IFC, meshes), the geometric primitives (OBB, PCA, surface descriptors), the statistical / machine-learning tools (dimensionality reduction, clustering algorithms, cluster-quality metrics), and the modern foundation models on which our text and visual modalities rest. Readers familiar with any block may skim it. The goal of the chapter is to fix notation and to make every formula that appears later self-contained.

2.1 Building Information Modeling and the IFC Schema

2.1.1 What BIM is, operationally

Building Information Modeling (BIM) is the practice of representing a built asset as a structured, semantically-typed collection of geometric objects with associated properties, relationships, classifications, and quantities. A modern BIM model is therefore not "a 3D drawing" but a graph whose nodes are typed entities and whose edges carry composition, aggregation, voiding, hosting, and spatial-containment semantics. The dominant authoring tools (Autodesk Revit, Bentley OpenBuildings Designer, Graphisoft ARCHICAD, Allplan, etc.) each maintain their own internal product model; they export to a neutral schema for exchange.

2.1.2 The IFC schema

Industry Foundation Classes (IFC) is the open ISO-standardised exchange format (ISO 16739) for BIM, maintained by buildingSMART International. IFC4 and IFC4.3 are the relevant current releases. The schema is expressed in EXPRESS, instantiated as a STEP physical file (`.ifc`), and shipped to consumers along with optional secondary representations (`.ifcXML`, `.ifcZIP`, `.bcf`, etc.).

Conceptually, an IFC project is rooted at `IfcProject`, decomposes into spatial containers (`IfcSite`, `IfcBuilding`, `IfcBuildingStorey`, `IfcSpace`), and populates them with `IfcBuildingElement` subtypes:

- **Structural:** `IfcBeam`, `IfcColumn`, `IfcMember`, `IfcSlab`, `IfcWall`, `IfcFooting`, `IfcPile`, `IfcRoof`, `IfcPlate`, `IfcCurtainWall`.
- **Architectural:** `IfcDoor`, `IfcWindow`, `IfcRailing`, `IfcStair`, `IfcRamp`, `IfcCovering`.
- **Equipment and furniture:** `IfcFurniture`, `IfcFurnishingElement`, `IfcLightFixture`, `IfcSanitaryTerminal`, etc.
- **Building services (MEP):** `IfcFlowTerminal`, `IfcDistributionFlowElement`, `IfcDuctSegment`, `IfcPipeFitting`, ...
- **Catch-alls:** `IfcBuildingElementProxy`, `IfcBuildingElementPart`.

Each entity carries a global identifier (`GlobalId`, a 22-character encoded GUID), a placement (`IfcLocalPlacement`), one or more shape representations (`IfcShapeRepresentation`), and a property-set bag (`IfcPropertySet`). Geometry can be expressed as parametric (extrusion, sweep, CSG), boundary representation (B-Rep), or tessellated (`IfcTriangulatedFaceSet`) form. For machine learning on shape, the common workflow is to evaluate the parametric/B-Rep geometry through an open-source kernel (we use `ifcopenshell` with its `geom` iterator) and obtain a triangulated mesh in world coordinates.

2.1.3 Why IFC types are noisy labels

The same physical object can legitimately appear under different `Ifc*` types in two submissions of the same building, for three structural reasons:

1. **Authoring-tool mapping tables.** The IFC export module inside each authoring tool ships a default Revit-family $\rightarrow$ IFC-type mapping. The user is free to override it on a per-element basis. A diagonal bracing element or a truss web member illustrates the problem cleanly: the schema admits two valid choices, `IfcBeam` (the default for Revit's Structural Framing category) and `IfcMember` (the schema-canonical type for slender load-carrying members whose orientation is not constrained to horizontal), and a custom family with no configured mapping falls through to `IfcBuildingElementProxy`. The same physical element therefore appears under three different IFC types across three submissions, with no schema-level signal that flags the inconsistency.
2. **Near-synonym entities.** IFC defines pairs such as `IfcWall` vs. `IfcWallStandardCase`, where the latter is the restricted subtype for axis-extruded walls. Many exporters use the two interchangeably.

3. **Catch-all types.** `IfcBuildingElementProxy` is explicitly the schema's escape hatch for "anything the exporter could not classify". `IfcBuildingElementPart` is a container for sub-components of an aggregate element. Both contain no geometric identity: they group objects by what they are *not*, rather than what they are.

For this reason, we treat IFC type as a noisy supervisory signal: it is useful for evaluation (macro purity, per-type recall) and for sanity checks, but it is not the ground truth of the reusable-component library we are mining.

2.2 3D Mesh Representation

A triangular mesh $\mathcal{M}$ is a 2-tuple $(\mathbf{V}, \mathbf{F})$ where $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^3 \mid i = 1, \ldots, n\}$ is the vertex set and $\mathbf{F} \subset \{(i,j,k) \mid i,j,k \in [1,n], i \neq j \neq k\}$ is the face set. We additionally work with derived quantities:

- Face area: $A_f = \frac{1}{2}\|(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)\|$ for $f = (i,j,k)$.
- Face normal: $\mathbf{n}_f = \frac{(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)}{\|(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)\|}$.
- Total surface area: $A = \sum_f A_f$.
- Surface centroid: $\mathbf{c}_S = \frac{1}{A} \sum_f A_f \mathbf{c}_f$ where $\mathbf{c}_f$ is the face centroid.
- Volume (closed orientable mesh): $V = \frac{1}{6} \sum_f \mathbf{v}_i \cdot (\mathbf{v}_j \times \mathbf{v}_k)$ (signed).

The closed-mesh volume formula is included for completeness: BIM meshes are frequently non-closed (open slab soffits, uncapped extrusions, t-junctions) so the divergence-theorem volume is not reliable on this corpus. The volume feature used throughout this thesis is therefore the OBB volume $e_1 \cdot e_2 \cdot e_3$ (Section 4.3.1), not `mesh.volume`.

2.3 Axis-Aligned and Oriented Bounding Boxes

2.3.1 AABB

The Axis-Aligned Bounding Box (AABB) of $\mathbf{V}$ is the smallest box whose faces are aligned with the world axes:

$$\mathrm{AABB}(\mathbf{V}) = \prod_{a \in \{x,y,z\}} \left[\min_i v_{i,a},\ \max_i v_{i,a} \right]. \tag{2.1}$$

It is trivial to compute but reflects the world coordinate system, not the object's own geometry. A unit cube or a square-cross-section element rotated by 45° in one coordinate plane has an AABB whose volume is exactly $2\times$ its true OBB volume. For the case that matters in BIM, a slender beam of object-frame extents (L, w, w) with $L \gg w$ rotated by 45° about an axis perpendicular to its length, the ratio grows with the slenderness and is approximately $L/(2w)$: an order of magnitude larger than the cube case for any realistic beam. Axis alignment is therefore not a constant-factor mismatch but a slenderness-dependent one, which matters in particular for structural framing.

2.3.2 OBB

The Oriented Bounding Box (OBB) is an oriented bounding box aligned to the object's own geometry rather than the world axes. We compute it with `trimesh.bounds.oriented_bounds`, which returns a near-minimum-volume box: it forms the convex hull of the vertex set, enumerates each hull face's outward normal as a candidate principal axis, applies rotating calipers in the plane perpendicular to that normal to obtain the minimum-area bounding rectangle, and selects the candidate of smallest resulting volume. This is distinct from a PCA-of-vertices "OBB", which we use separately for the engineered orientation features in Chapter 4; the PCA-aligned box is in general *not* minimum-volume, while the hull-face-enumeration approach is much closer to the true optimum. The returned extents are sorted as (e_1, e_2, e_3) with $e_1 \geq e_2 \geq e_3$. Figure 2.1 contrasts the axis-aligned and oriented boxes on a slender element, making the wasted-volume effect discussed above concrete.

The OBB is the natural reference frame for most structural BIM elements, which have a clearly dominant geometric direction (the beam axis, the wall normal, the slab plane). Elements with near-isotropic geometry or rotational symmetry (square footings, square columns, ceiling-mounted light fixtures, four-fold-symmetric sanitary terminals) have ambiguous principal directions and we treat the resulting extents as such. Throughout this thesis the four baseline OBB features are:

$$L = e_1 \qquad \text{(length, longest extent)} \tag{2.2}$$

$$C = e_2 \cdot e_3 \qquad \text{(cross-section area)} \tag{2.3}$$

$$V = e_1 \cdot e_2 \cdot e_3 \qquad \text{(OBB volume)} \tag{2.4}$$

$$N = |\mathbf{V}| \qquad \text{(vertex count)} \tag{2.5}$$

These four numbers are the IFCNet-style baseline against which all the feature-engineering progression of Chapter 4 will be benchmarked.

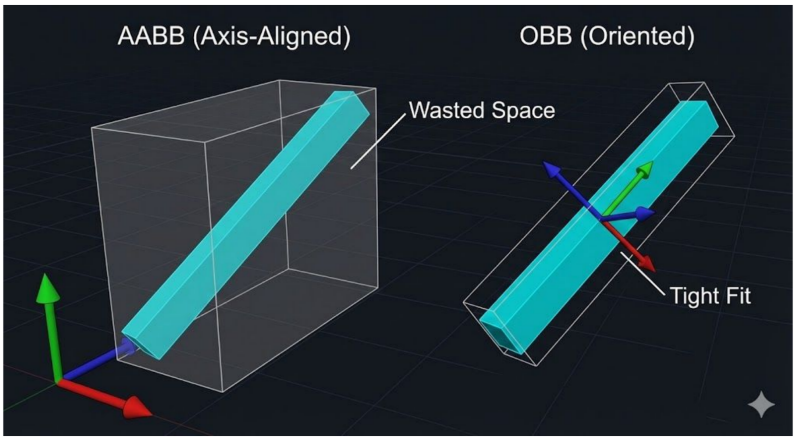


Figure 2.1: Axis-aligned (AABB) versus oriented (OBB) bounding box for a slender element. The world-aligned box leaves a large wasted-volume margin around a diagonally-placed object, whereas the OBB aligns to the element's own principal axes for a tight fit. As derived above, the wasted-volume ratio is not a constant factor but grows with slenderness, which is why the OBB is used as the geometric reference frame throughout this thesis.

2.4 Principal Component Analysis on Vertex Clouds

Given vertices $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ with $\mathbf{v}_i \in \mathbb{R}^3$, we centre them at the centroid $\bar{\mathbf{v}} = \frac{1}{n}\sum_i \mathbf{v}_i$ and form the $n \times 3$ centred matrix $\mathbf{C}$. The sample covariance

$$\boldsymbol{\Sigma} = \frac{1}{n-1}\mathbf{C}^\top \mathbf{C} \in \mathbb{R}^{3\times 3} \tag{2.6}$$

matches the convention used by `sklearn.decomposition.PCA`, which is the implementation we invoke. The factor $\frac{1}{n-1}$ versus $\frac{1}{n}$ only rescales the eigenvalues by a constant; the eigenvectors and the eigenvalue ratios we read off below are unchanged. $\boldsymbol{\Sigma}$ is symmetric positive semi-definite, so its eigendecomposition $\boldsymbol{\Sigma} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$ gives non-negative eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ with orthonormal eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$. The eigenvectors are sign-ambiguous; the features below are all invariant to this sign.

2.4.1 PCA-axis orientation features

The unsigned dot product of a principal axis with the world z-axis gives a measure of how vertical or horizontal that direction is. The absolute value resolves the sign-ambiguity of the eigenvector. We define, for the longest and the thinnest principal

directions,

$$\mathrm{pc}_k^z = |\mathbf{q}_k \cdot \mathbf{e}_z|, \quad k \in \{1, 3\}. \tag{2.7}$$

$\mathrm{pc}_1^z \to 1$ indicates a vertically-oriented longest axis (columns, posts, vertical pipes); $\mathrm{pc}_1^z \to 0$ indicates a horizontally-oriented longest axis (beams, horizontal pipes). Symmetrically, $\mathrm{pc}_3^z \to 1$ indicates a vertical thinnest axis (slabs, footings, roofs, coverings); $\mathrm{pc}_3^z \to 0$ indicates a horizontal thinnest axis (walls, plates). These two features are central to the orientation stage of our engineered feature progression (Section 4.3).

2.4.2 Cross-section aspect ratio

The two minor PCA eigenvalues describe the dispersion of vertices in the plane perpendicular to the principal axis, i.e. the cross section. Their square-root ratio

$$\text{cs\_aspect\_ratio} = \sqrt{\lambda_3 / \lambda_2} \tag{2.8}$$

is bounded in $[0, 1]$. Values near 1 indicate a square / circular cross section (columns, square posts), values near 0 indicate a flat cross section (plates, slabs, walls, coverings).

2.5 Surface-Normal Statistics

For each face f with unit normal $\mathbf{n}_f$ and area A_f we construct an area-weighted distribution of normal orientations on the sphere S^2.

2.5.1 Horizontal-face fraction

The simplest statistic is the fraction of total area whose normal is within $\cos^{-1}(0.9) \approx 25.84^\circ$ of the world z-axis:

$$\text{horiz\_frac} = \frac{1}{A} \sum_f A_f \cdot \mathbb{1}\Big[\, |\mathbf{n}_f \cdot \mathbf{e}_z| > 0.9 \,\Big]. \tag{2.9}$$

horiz_frac $\to 1$ indicates a flat-topped element (slab, roof, covering); horiz_frac $\to 0$ indicates a primarily vertical-walled element (beam, column, wall).

2.5.2 Normal entropy

A richer surface descriptor is the Shannon entropy of the area-weighted normal histogram on an icosphere of $B = 42$ bins (an icosphere at subdivision level 1). Let $p_b = \frac{1}{A} \sum_{f \in b} A_f$ be the area fraction in bin b. Then

$$\text{normal\_entropy} = -\frac{1}{\log B} \sum_{b=1}^{B} p_b \log p_b. \tag{2.10}$$

The $1/\log B$ normalisation puts the value in $[0, 1]$: low entropy indicates an axis-aligned extruded element with only a few unique face orientations (slab, plate), high entropy indicates a freeform / detailed element (furniture, light fixture, MEP fitting).

2.6 Dimensionality Reduction

We work with three classes of reducer.

2.6.1 Identity

A no-op reducer; used when the input dimensionality is already small (e.g. the 9-d engineered geometric block).

2.6.2 PCA

Given a centred data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, PCA computes the singular value decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and projects $\mathbf{X}$ onto the leading r right-singular vectors:

$$\mathbf{X}_{\text{red}} = \mathbf{X}\,\mathbf{V}_{:,1:r} \in \mathbb{R}^{N \times r}. \tag{2.11}$$

PCA is linear, deterministic, and preserves global variance structure. It tends to compress all directions of variance uniformly; very fine local neighbourhood structure may be smeared.

2.6.3 UMAP

Uniform Manifold Approximation and Projection (UMAP) [1] is a non-linear neighbourhood-preserving manifold-learning algorithm. It builds a fuzzy-simplicial neighbour graph in the input space (parametrised by `n_neighbors` and `min_dist`), then optimises a low-dimensional embedding that preserves the cross-entropy between the high- and low-dimensional fuzzy graphs. Compared with PCA, UMAP preserves *local* neighbourhood structure much more aggressively, which we find empirically to be valuable for downstream HDBSCAN clustering on text and visual embeddings (Section 5.4, Section 5.5).

We use a fixed configuration of `n_neighbors` $= 30$, `min_dist` $= 0.0$ across all UMAP reductions, sweeping only the output dimensionality $r \in \{4, 8, 16, 64\}$.

2.7 Clustering Algorithms

We evaluate four clustering families.

2.7.1 K-Means

Given N points $\mathbf{X} = \{\mathbf{x}_i\}$ and K centroids $\boldsymbol{\mu}_k$, K-Means [2] minimises

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2. \tag{2.12}$$

We use the standard `scikit-learn` implementation (`sklearn.cluster.MiniBatchKMeans`, `batch_size`=1024, `n_init=auto`), re-fit over 10 random seeds, and report the median where relevant.

K-Means assumes (a) the number of clusters is known, (b) clusters are isotropic, and (c) clusters are similar in size. None of these hold for the BIM library mining problem, but K-Means remains a useful baseline and, as Chapter 5 will show, at *matched* k it is competitive with HDBSCAN on partition purity; HDBSCAN's advantage is the automatic selection of k.

2.7.2 Bisecting K-Means

Bisecting K-Means is a divisive top-down variant: it starts with all points in one cluster, then repeatedly picks the worst cluster (by sum-of-squared-errors) and splits it into two with binary K-Means. After $K - 1$ bisections, K clusters remain. It tends to produce hierarchical structures with fewer "stuck" centroids but pays a modest purity cost on the data here.

2.7.3 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) assumes the data are i.i.d. samples from

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.13}$$

fit by Expectation-Maximization (EM). We evaluate both diagonal and full covariance matrices. GMM gives *soft* assignments (posterior responsibilities) which we hard-assign for evaluation. With full covariance, GMM generalises K-Means to anisotropic clusters and often does slightly better in pure-clustering metrics, but the cost is more parameters per cluster.

2.7.4 HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [3] is a density-based hierarchical clustering algorithm that extends DBSCAN by

avoiding the choice of a single global density threshold. The algorithm proceeds in five steps:

1. Compute mutual reachability distances $d_{\text{mreach}}(\mathbf{x}_i, \mathbf{x}_j) = \max\{\text{core}_k(\mathbf{x}_i), \text{core}_k(\mathbf{x}_j), d(\mathbf{x}_i, \mathbf{x}_j)\}$, where core_k is the distance to the k-th nearest neighbour.
2. Build the minimum spanning tree on the mutual reachability graph.
3. Extract the cluster hierarchy as a dendrogram of the MST by progressively raising the edge-weight threshold.
4. Condense the tree by collapsing splits that produce clusters smaller than `min_cluster_size` ($= mcs$).
5. Select flat clusters from the condensed tree by either *Excess of Mass* (`eom`, which globally maximises cluster stability) or *leaf* (`leaf`, which selects every leaf as a cluster, yielding finer granularity).

The two hyperparameters are `min_cluster_size` and `min_samples` ($= ms$, controlling the local density threshold). Points that lie outside any condensed cluster are labelled *noise* (cluster -1). The `force_assign` flag, when enabled, is a post-processing step that assigns each noise point to the nearest cluster centroid by Euclidean distance in the input embedding (it is our own wrapper around the `hdbscan` library output, not a stock parameter). We use it throughout to obtain a complete, no-noise partition suitable for downstream catalog and retrieval applications, and report the noise-preserving variant separately as a diagnostic.

HDBSCAN does not require pre-specifying K; the effective K is implicitly chosen by the data density. For the BIM library mining problem, where the true number of reusable components is unknown and clearly differs from the 17 IFC label classes, this is the crucial property: the algorithm answers *both* "how many clusters?" *and* "which point goes where?" from a single density estimate.

2.8 Cluster Quality Metrics

2.8.1 Internal metrics

Silhouette score. For point i,

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.14}$$

with $a(i)$ the mean distance to other points in its cluster and $b(i)$ the mean distance to the nearest other cluster. The overall score is $\bar{s} = \frac{1}{N}\sum_i s(i) \in [-1, 1]$. Higher is better; values > 0.5 indicate strong structure.

Davies–Bouldin Index.

$$\mathrm{DB} = \frac{1}{K}\sum_{k=1}^{K} \max_{j \neq k} \frac{s_k + s_j}{d(c_k, c_j)}, \tag{2.15}$$

where s_k is the mean intra-cluster distance and $d(c_k, c_j)$ is the centroid distance. Lower is better.

2.8.2 External metrics (using labels)

Purity / Macro purity. Given a partition $\{C_k\}$ and ground-truth classes $\{T_y\}$, each cluster is assigned the class it most overlaps,

$$\ell(k) = \arg\max_y |C_k \cap T_y|, \tag{2.16}$$

which turns the partition into a classifier: every object is predicted the dominant class of its cluster. *Overall purity* is the accuracy of that classifier,

$$\mathrm{purity} = \frac{1}{N}\sum_k |C_k \cap T_{\ell(k)}| = \frac{1}{N}\sum_k \max_y |C_k \cap T_y|. \tag{2.17}$$

The *per-type purity* of class y is the fraction of its objects that fall in a cluster assigned to y, that is, the recall of class y under ℓ,

$$\mathrm{pur}_y = \frac{1}{|T_y|}\sum_{k\,:\,\ell(k)=y} |C_k \cap T_y|, \tag{2.18}$$

which is 0 for a class that is the dominant label of no cluster. *Macro purity* averages this uniformly over classes, while overall purity is the same average weighted by class size:

$$\mathrm{macro\_purity} = \frac{1}{|Y|}\sum_{y \in Y} \mathrm{pur}_y, \qquad \mathrm{purity} = \sum_{y \in Y} \frac{|T_y|}{N}\,\mathrm{pur}_y. \tag{2.19}$$

Macro purity is our headline number: it gives every IFC type the same vote, so a clustering that does well on `Furniture` (n=258) but ignores `Member` (n=50) is penalised. By construction it rises with the cluster count k and reaches 1 when every object forms its own cluster; it is therefore read together with the fixed-k comparison of Section 5.6.5, not as a k-independent score.

Adjusted Rand Index.

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}. \tag{2.20}$$

ARI corrects the raw Rand Index for chance agreement and ranges in $[-1, 1]$ (with 0 indicating chance-level agreement and 1 identical partitions). We use ARI to quantify stability under bootstrap resampling.

2.9 Vision Encoders for 3D-Object Renders

We pass nine canonical-view renders per object through three modern visual foundation models. Each is briefly described below.

2.9.1 DINOv3

DINO [4] is a self-supervised Vision Transformer trained on ImageNet via student/teacher self-distillation without labels. DINOv3 [5] (the version used here) extends to larger scales and stronger augmentations and adopts register tokens [6] to suppress high-norm attention artefacts. We use the ViT-L/16 backbone via the public `dinov3_vitl16_pretrain_lvd1689m` checkpoint and take the `CLS` token of the final layer as the 1024-d image embedding.

2.9.2 SigLIP2

SigLIP [7] replaces the softmax contrastive loss of Contrastive Language-Image Pre-training [8] with a pairwise sigmoid loss, allowing the loss to be evaluated independently per image–text pair. SigLIP2 [9] is the second-generation release with improved zero-shot performance. We use the `google/siglip2-large-patch16-512` backbone and take the image-side embedding (1024-d).

2.9.3 DuoDuo CLIP

DuoDuo CLIP [10] is a CLIP variant trained for 3D-shape understanding via a multi-view contrastive objective and "object-centric" image augmentations. Unlike DINOv3 and SigLIP2, DuoDuo CLIP is a native multi-view model: its `encode_image` call accepts the whole set of viewpoint renders at once and fuses them inside its own forward pass. We therefore pass all nine views together and take its single 512-d output directly, without any external pooling.

2.9.4 Embedding aggregation

For DINOv3 and SigLIP2 we compute nine per-view embeddings and mean-pool them into a single per-object embedding. Mean-pooling is the simplest view-invariant aggregator; it matches the original SigLIP and DINOv3 evaluation protocols and avoids confounding the encoder comparison with the aggregation choice. DuoDuo CLIP, as noted above, fuses the nine views inside its own forward pass, so for that encoder no external pooling is applied.

2.10 Vision–Language Models for Object Description

We additionally generate *text* descriptions of each object by prompting a Vision-Language Model (VLM). Two ingredients matter.

2.10.1 Gemini 3.1 Flash Lite as the VLM

Gemini 3.1 Flash Lite (`gemini-3.1-flash-lite-preview`) is the multimodal generative model used as the description producer. Given a multi-image prompt (the nine views) together with the object's measured geometry (its OBB aspect ratio, cross-section thinness and PCA-axis orientation) and a system instruction asking for a single compact description, it returns a one-line tag string of the form `[thinness]`, `[shape-class]`, `[aspect-ratio]`, `[profile]`, `[visual-cue]` followed by one to five distinguishing keywords. The instruction requires the geometric tags to be consistent with the supplied measurements, and forbids both material/colour terms (not recoverable from stripped geometry) and IFC class names. The prompt evolves across seven versions (Chapter 4, Section 4.5).

2.10.2 Gemini Embedding 2 as the text encoder

The one-line tag string is embedded with the Gemini Embedding 2 model, which produces a 768-d sentence embedding. This embedding is what enters the clustering pipeline. Gemini Embedding 2 is used as a current state-of-the-art general-purpose text embedder; we do not claim superiority over a specific open-source alternative, and the contribution of the text modality is evaluated directly against the geometric and visual modalities in Chapter 5.

2.10.3 Why two-stage VLM $\rightarrow$ text-encoder?

A purely image-side embedding (DINOv3, SigLIP, DuoDuo) and a purely text-side embedding via VLM describe very different things: the former is the appearance, the

latter is the named shape/material/function. We hypothesise, and Chapter 5 confirms, that the two are partially redundant but partially complementary, with text recovering identity-ambiguous types (`IfcMember`, `IfcPlate`) where image features alone struggle.

2.11 Statistical Tests

For per-feature discriminative power we use one-way analysis of variance (the F-test from `scipy.stats.f_oneway`) across the IFC-type groups. Because the benchmark has $N = 1{,}700$ objects and $k = 17$ classes ($\mathrm{df}_{\text{between}} = 16$, $\mathrm{df}_{\text{within}} = 1{,}683$), any non-noise feature reaches extreme statistical significance, so the p-value alone is uninformative. We therefore rank features by the effect size, eta-squared $\eta^2 = \mathrm{SS}_{\text{between}}/\mathrm{SS}_{\text{total}}$ — equivalently $\eta^2 = F\,\mathrm{df}_{\text{between}}/(F\,\mathrm{df}_{\text{between}} + \mathrm{df}_{\text{within}})$ — the share of total variance explained by the type grouping. Every feature clears significance by a wide margin ($p_{\text{holm}} \ll 10^{-100}$ after Holm correction across the panel), so the correction changes nothing; the substantive comparison is by η^2.

2.12 Summary

This chapter has fixed (i) the data domain (BIM, IFC, mesh representations), (ii) the geometric primitives (OBB, PCA, surface-normal statistics) that we will compose into engineered features, (iii) the dimensionality reducers (PCA, UMAP), (iv) the clustering algorithms (K-Means, Bisecting K-Means, GMM, HDBSCAN), (v) the quality metrics (silhouette, DB, CH, purity, macro purity, ARI), and (vi) the foundation models (DINOv3, SigLIP2, DuoDuo, Gemini) that produce our visual and textual modalities. The next chapter positions this toolbox against the prior art.

3 Related Work

This chapter positions the thesis against four streams of prior art: (i) machine learning on BIM/IFC data, (ii) handcrafted and learned 3D shape descriptors, (iii) multi-modal and language-grounded 3D understanding, and (iv) the emerging line of agentic / MCP-driven tools for IFC. We close with a synthesis that motivates the specific multi-modal, unsupervised, library-mining angle of the present work.

3.1 Machine Learning on BIM and IFC Data

3.1.1 The "Generic Class Problem"

A consistent finding in the BIM-quality literature is that `IfcBuildingElementProxy` and similar catch-all entities are used far in excess of what the schema intends. Noardo et al. [11] inspect real-world IFC submissions and document substantial quality and consistency variation across authored models, including over-use of generic catch-all entities through export-tool defaults or user error. Koo & Shin [12] formalise this as a novelty-detection problem and show that one-class SVMs on geometric features identify a substantial fraction of mislabelled elements.

Beyond classification correctness, several works study compliance and quality checking. Ontology-driven compliance checking (e.g. via SWRL rule sets layered on the IFC ontology) provides one direction [13, 14], while more recent work fuses LLMs with deep models for richer compliance reasoning [15]. Both directions are *deductive*: rules are written by experts and matched against the model. The present thesis is the dual: it is inductive, surfacing structure from unlabelled data without rules.

3.1.2 Supervised IFC entity classification

The dominant response to the generic-class problem is to predict the correct IFC type from geometry with a supervised classifier. **IFCNet** [16] is a large-scale benchmark for this task: 19,613 single-entity IFC objects across 65 classes, together with a class-balanced 20-class subset (IFCNetCore, 7,930 objects) and a fixed train/test split; its baselines are MVCNN [17], DGCNN [18] and MeshNet [19]. **SpaRSE-BIM** [20], by the same group, replaces these with sparse-convolutional networks that trade a small

accuracy drop relative to the best multi-view model (macro-F1 ≈0.82 versus MVCNN's 0.87) for substantially lower inference cost. **IFCGeoNet** [21] combines multi-view rendering with a heterogeneous graph encoder.

An earlier, more classical line is data-driven IFC-object classification. Wu and Zhang [22] derive an iterative rule-plus-learning algorithm that maps IFC objects into predefined categories to repair interoperability loss. Koo et al. [23] use support-vector machines on geometric metrics for semantic-integrity checking, and later show with 3D geometric deep neural networks (MVCNN, PointNet) that even *sub-types* of walls and doors can be recovered automatically [24]. A separate random-forest study on prefabricated components [25], together with the recent work of Tang et al. [26] which classifies 8,715 BIM objects under the Uniclass standard with a random forest over thirteen IFC-derived features (semantic, spatial and dimensional), are the closest supervised analogues to the descriptor-plus-Random-Forest ceiling we report in Chapter 5. Abualdenien and Borrmann [27] apply the same recipe to a different target, classifying the *Level of Geometry* of building elements with tree-based ensemble models over sixteen geometric-complexity features, confirming that compact handcrafted descriptors carry enough signal for ensemble classifiers to label building elements reliably.

A third strand encodes the object as a graph and applies geometric deep learning. Collins et al. assess IFC classes with graph convolutional networks over different mesh-graph encodings [28] and extend the idea to "semantic healing" of mislabelled design models, transferring the learned shape encoding to scan-to-BIM segmentation [29]; Luo et al. [30] fuse a geometric branch with a relational graph branch for the same classification task. Beyond element identity, Yu et al. [31] apply a multi-view vision transformer to classify *clashes* between disciplines into a penetration taxonomy.

Every one of these works shares one assumption: the IFC label set *is* the target. The classifier is trained to reproduce it and success is measured as agreement with it. The present thesis takes the dual position. It treats the label set as a noisy reference, clusters without supervision, and evaluates at a granularity ($k \gg |Y|$) that the schema itself cannot express.

3.1.3 Unsupervised structure recovery from BIM

Jin et al. [32] provide an early proof-of-concept for unsupervised graph-based learning on BIM data, focusing on `IfcSpace` rooms rather than structural components. Wang & Ying [33] embed multimodal BIM models with graph representation learning; their finding that semantic text embeddings beat raw point-cloud geometry on noisy authored data is a key signal that pushed us towards a multi-modal rather than point-cloud-only design.

The current thesis differs from these in three respects: (a) we operate at the per-element level rather than at the room/space level, (b) we mine library structure (subtypes within IFC labels) rather than classifying into existing labels, and (c) we evaluate against per-type macro purity at $k \gg |Y|$, surfacing structure beyond the schema grain. The shape-mining literature on `ShapeNet/PartNet` explores a related question (subcategory recovery from clean catalogues); we deliberately work on noisy, federated BIM submissions where the labels themselves are unreliable, which the `ShapeNet`-trained encoders we use as feature extractors were not designed for.

3.2 3D Shape Descriptors

3.2.1 Handcrafted descriptors

The Osada D2 shape distribution [34] is a foundational descriptor: histograms of pairwise Euclidean distances between random surface points. It captures global shape without requiring correspondence, is rotation-invariant by construction and scale-invariant under appropriate normalisation, and remains a strong baseline. We do not use D2 directly but our `normal_entropy` and OBB-derived statistics are in the same spirit: simple, deterministic, interpretable surface descriptors.

PCA-based dimensionality descriptors [35, 36] (linearity, planarity, scattering) are a staple of LiDAR point-cloud semantic segmentation; we draw on the same eigen-decomposition of the vertex cloud, but, as Chapter 4 shows, our engineered geometry instead uses orientation and cross-section-ratio descriptors derived from the principal axes. Fourier shape descriptors [37], spherical-harmonics representations [38], light-field descriptors [39] and the related family of D3/D4 statistics are well-covered in the survey of Tangelder & Veltkamp [40].

Local descriptors such as **FPFH** [41], **SHOT** [42] and spin images [43] describe point neighbourhoods rather than whole objects, and are crucial for registration and partial-matching tasks. They are not directly applicable to library mining where the object-level signal is what matters.

3.2.2 Deep learned descriptors

PointNet [44] and **PointNet++** [45] are the architectural lineage from which most modern point-cloud encoders descend: per-point MLP plus a permutation- invariant aggregator. **DGCNN** [18] introduces dynamic graph construction in feature space. **PointTransformer** [46] and **PCT** [47] adapt the transformer architecture to point clouds.

For *multi-view* learning on 3D shapes, **MVCNN** [17] renders the object from several viewpoints and applies a 2D CNN; this is the historical antecedent of our nine-viewpoint

visual-encoder pipeline, with the difference that we substitute a pretrained Vision Transformer (DINOv3, SigLIP2) for the trained-from-scratch CNN and aggregate by mean pooling rather than view-pooling layers.

3.3 Vision Foundation Models for 3D Understanding

3.3.1 Self-supervised image encoders

DINO [4] demonstrated that self-distillation without labels produces visual features with strong *k*-NN and linear classification performance. **DINOv2** [48] scaled this to 142M images and produced features that work well out-of-the-box for retrieval, depth, and segmentation. Register tokens, which absorb high-norm artefacts in the attention maps and yield cleaner dense features, were introduced separately by [6] and first shipped in DINOv2-with-registers. **DINOv3** [5] extends the family with Gram anchoring, a much larger training corpus, and post-hoc high-resolution adaptation. In our experiments we use the ViT-L/16 backbone of DINOv3.

3.3.2 Vision–language contrastive models

CLIP [8] aligns image and text in a shared embedding space via softmax-contrastive learning on 400M web pairs. **SigLIP** [7] swaps the softmax for a sigmoid loss, decoupling the per-pair loss from the batch size. SigLIP2 [9] is the v2 release used here.

For 3D specifically, **DuoDuo CLIP** [10] adapts CLIP to render-image inputs with a multi-view contrastive objective. **OpenShape** [49] and **ULIP-2** [50] train shared embeddings of text, image, and point cloud, enabling zero-shot 3D classification. These are direct intellectual ancestors of our visual-text fusion, but operate at the ShapeNet/Objaverse domain and are not BIM-aware.

3.3.3 Vision–language models as describers

A separate use of multimodal models is as *describers* rather than encoders. **LLaVA** [51], **Qwen-VL** [52], and the Gemini family [53] take images + a text prompt and produce free-form text. Recent shape- description work (e.g. **Cap3D** [54]) generates captions for ShapeNet/Objaverse via an LLM ensemble. We follow the same pattern but constrain the LLM to a JSON schema with explicit shape, material-class, and function-class fields, and we sweep seven prompt versions to surface IFC-aware vocabulary.

3.4 Multi-Modal Fusion

Multi-modal fusion has been extensively studied in vision–language retrieval, action recognition, and medical imaging. It is conventionally grouped into "early" (input-level), "intermediate" (feature-level), and "late" (decision-level) fusion [55], and the relative merits of early versus late fusion have been studied empirically [56]. These map cleanly to our F1a (intermediate fusion via concatenation), F5 (intermediate fusion via UMAP-on-concatenation), and F4 (late fusion via Borda) recipes. Canonical Correlation Analysis [57, 58] provides an intermediate-fusion alternative we evaluate but do not deploy.

Our own recipe, "each modality reduced individually, then concatenated and clustered", sits in the mid-fusion family above; we are not aware of a direct precedent that applies it to *unsupervised* per-element BIM library mining. One recurring lesson from this literature, which our results in Chapter 5 echo quantitatively, is that "adding a modality" is not free: two modalities derived from overlapping signal (here, image renders and VLM descriptions *of* those same renders) can be partially redundant, so the marginal lift of stacking modalities is bounded by their independence rather than by the number of streams fused.

3.5 Agentic and MCP-Driven IFC Tools

A very recent line of work uses the Model Context Protocol (MCP) [59] to drive LLM agents over IFC data. **MCP4IFC** [60] is an LLM-driven *generative* building-design framework: it couples an MCP server wrapping `ifcopenshell` with code generation and retrieval-augmented prompting so that an agent can author and modify IFC elements from natural-language instructions, rather than merely read an existing model. The thesis title, *Mining of Reusable Component Libraries through Unsupervised Multi-Modal Clustering*, deliberately stays out of the open-vocabulary retrieval space that an MCP integration would enable. We position MCP4IFC as a complementary, downstream layer: once the clusters and the catalog exist, exposing them through MCP is a near-term deployment step rather than a research contribution.

3.6 Synthesis and Gap

Three observations emerge from the literature.

1. **Supervised IFC classification is well-studied** (IFCNet, SpaRSE-BIM, IFCGeoNet, RF-on-prefab) but assumes the label set is the truth. None of these works address

the *schema-grain* problem: the fact that two physically identical objects can sit in different IFC types, and that two physically distinct sub-categories share one label.

2. **3D shape descriptors and learned encoders** are mature technologies, but were developed and evaluated on ShapeNet / ModelNet / Objaverse, where labels are exhaustive and clean. BIM data has noisier labels, more dataset-specific texture artefacts, and a strong tail of detailed CAD objects (furniture, MEP) that fall outside the typical evaluation distributions.

3. **Multi-modal fusion on BIM data is rare.** The few works that combine geometry + text (Wang & Ying) operate at the project level or via graph embedding; per-element multi-modal clustering for library mining, on noisy federated submissions, is to our knowledge unaddressed in the BIM-ML literature.

The gap into which this thesis falls is therefore: *a reproducible, per-element, unsupervised, multi-modal mining pipeline that combines handcrafted geometry, VLM-derived text descriptions, and modern visual foundation models, evaluated against a noisy IFC-type ground truth, with a downstream library-application layer (catalog and retrieval).* The remaining chapters describe how we fill that gap, and Chapter 5 returns to the modality-redundancy issue raised above with quantitative evidence.

4 Methodology

This chapter describes the end-to-end pipeline of the thesis: from raw IFC submissions to a clustered library catalog and a fused retrieval embedding. The presentation follows the same order as the data flow. Section 4.1 gives the high-level view; Section 4.2 the dataset construction and the representative-subsampling strategy; Section 4.3 the four-stage geometric feature-engineering progression; Section 4.4 the 9-viewpoint render pipeline; Section 4.5 the seven-version prompt-engineering study for the text modality; Section 4.6 the three visual encoders; Section 4.7 the seven fusion recipes; Section 4.8 the clustering search space; Section 4.9 the evaluation protocol; and Section 4.10 the two downstream applications (catalog and retrieval).

4.1 Pipeline Overview

At a glance, the pipeline performs the following steps (Figure 4.1):

1. **IFC extraction.** 202 student-submission `.ifc` files are walked with `ifcopenshell`; each `IfcBuildingElement` subtype is evaluated to a triangulated mesh and stored as a `.obj` file indexed by its `GlobalId`. Yield: 128,883 meshes across 24 IFC types.
2. **Representative sub-sampling.** For each IFC type, we fit K-Means with up to $K = 100$ clusters on the 9-dimensional geometric feature vector (StandardScaler-standardised), then for each cluster centroid snap to the nearest real mesh. This produces $\sim$2,296 representative annotation candidates (some small types have < 100 objects; we keep all).
3. **Manual annotation and curation.** 1,849 of the 2,296 candidates receive hand annotation. We exclude two schema catch-all types (`IfcBuildingElementProxy`, `IfcBuildingElementPart`; 149 objects), then apply a type-merging map (`TYPE_MAP`) that collapses 22 source types into 17 evaluation classes (e.g. `IfcWall` + `IfcWallStandardCase` into `Wall`). Final benchmark: **1,700 objects across 17 classes**, no objects lost in the merge step.
4. **Geometric feature extraction.** Nine handcrafted features are computed per mesh: four log-OBB baseline plus five engineered (`pc1_z`, `pc3_z`, `cs_aspect_ratio`, `normal_entropy`, `horiz_frac`). See Section 4.3.

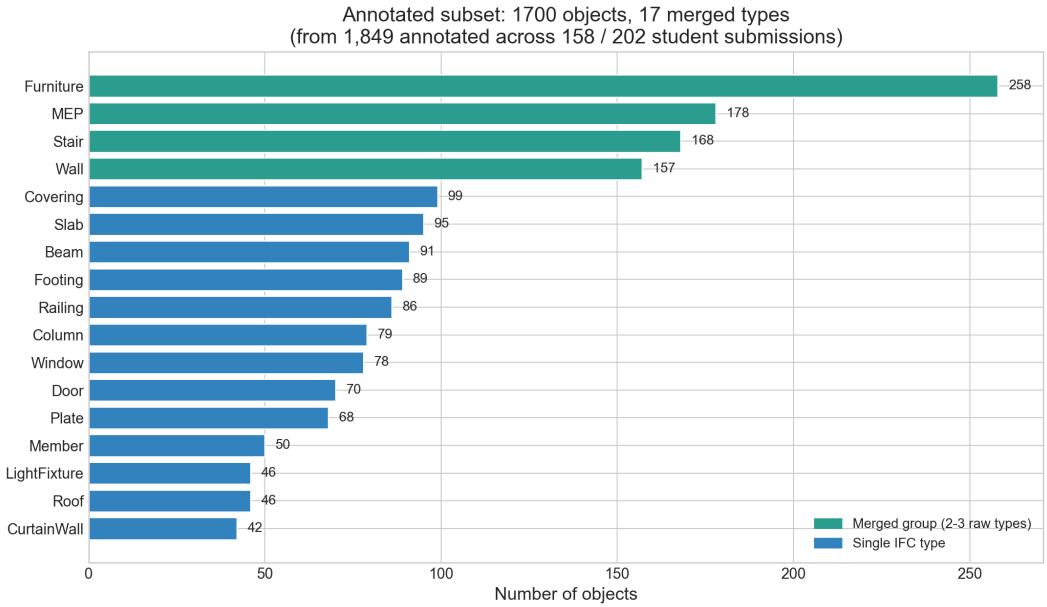


Figure 4.1: Dataset construction overview. 128,883 raw meshes extracted from 202 IFC submissions are reduced to 2,296 annotation candidates by per-type K-Means representative sub-sampling. After manual annotation, exclusion of schema catch-all types, and near-synonym type merging, the final benchmark contains **1,700 objects across 17 merged IFC classes**.

5. **9-viewpoint rendering.** Each mesh is centred and rendered from nine fixed viewpoints (`top`, `front`, `right`, `back`, `iso`, `left`, `front_right`, `back_left`, `bottom`).

6. **Text modality.** The nine views per object, together with the object's measured geometry, are sent to Gemini 3.1 Flash Lite with a system instruction asking for a single-line tagged description (thinness, shape class, aspect ratio, profile, visual cue, plus distinguishing keywords). The response string is embedded with Gemini Embedding 2 (768-d).

7. **Visual modality.** The nine coloured views are passed through three pretrained encoders: DINOv3 (1024-d), SigLIP2 (1024-d) and DuoDuo CLIP (512-d). For DINOv3 and SigLIP2 the nine per-view embeddings are mean-pooled into a single per-object vector; DuoDuo CLIP fuses the views internally and emits one vector directly. A colourless render variant is also produced as a robustness ablation.

8. **Per-modality dimensionality reduction.** Each modality is independently re-

duced through a sweep of reducers (identity, PCA $\in \{8, 16, 32, 64, 128\}$, UMAP $\in \{4, 8, 16, 64\}$), yielding a per-modality embedding.

9. **Fusion.** Seven fusion recipes (F1a, F1a_umap, F1b, F1b_varbal, F5, F5_varbal, CCA) combine the three modalities into a single fused embedding. See Section 4.7.
10. **Clustering.** K-Means, Bisecting K-Means, GMM and HDBSCAN are evaluated across the relevant hyperparameter grids.
11. **Evaluation.** Macro purity, overall purity, silhouette, Davies–Bouldin, per-type purity, and bootstrap ARI are computed.
12. **Downstream applications.** Auto-generated catalog and late-Borda retrieval.

The total compute envelope is a few hours on an Apple-Silicon laptop (M-series, 16 GB), excluding the one-time Gemini API descriptions (variable, $\sim$30 min).

4.2 Dataset Construction

4.2.1 Raw extraction

The corpus is the set of student IFC submissions of the TUM "Computational Methods in Building Engineering" course (2022–2025 cohorts, Prof. Borrmann's group). Each `.ifc` file is loaded with `ifcopenshell.open`, and the `geom.iterator` is configured with settings:

- `USE_WORLD_COORDS` = True
- `WELD_VERTICES` = True
- `INCLUDE_CURVES` = False
- `USE_PYTHON_OPENCASCADE` = False

For every `IfcProduct` descended from `IfcBuildingElement`, we obtain a triangulated mesh in world coordinates and write it to disk as an OBJ file indexed by its `GlobalId`. We additionally record one metadata entry per object: its `GlobalId`, `IfcType`, mesh filename, and source file.

Yield across the 202 submissions: 128,883 meshes across 24 `Ifc*` types.

4.2.2 Per-type representative sub-sampling

Manual annotation of ≈110,000 objects is infeasible. Random sampling would severely under-represent small IFC types and over-represent the most populous. We instead use a one-shot *K-Means-snap* representative-selection procedure:

1. For each IFC type t with population N_t, set $K_t = \min(100, N_t)$.
2. Fit K-Means with K_t clusters on a 9-dimensional shape-descriptor vector — the four log-OBB features plus five early shape descriptors (two OBB aspect ratios, a PCA planarity score, a face-area coefficient of variation and a convex-hull vertex ratio) — after standardisation, with `n_init`= 20. These descriptors are deterministic functions of the mesh, so they are available for every object before annotation. This exploratory descriptor set predates, and is independent of, the engineered feature progression of Section 4.3; here it serves only to spread the annotation pool across geometrically distinct objects.
3. For each cluster centroid $\boldsymbol{\mu}_k^{(t)}$, find the nearest *real* mesh:

$$\text{exemplar}_k^{(t)} = \arg\min_{\mathbf{x}_i \in t} \|\mathbf{x}_i - \boldsymbol{\mu}_k^{(t)}\|. \tag{4.1}$$

   Include this exemplar in the annotation pool.

This is a single-pass representative-selection scheme: snapping each K-Means centroid to its nearest real object yields a genuine exemplar per cluster while avoiding the $O(K \cdot N^2)$ pairwise-swap cost of iterative exemplar-refinement methods. The motivation is per-class *diversity coverage* at a fixed annotation budget: rather than over-sampling the populous types and losing the long tail, we want each IFC type to contribute its geometrically distinct exemplars. We measure this directly: for every merged class we fit `MiniBatchKMeans` with $K = 50$ on the full population's log-OBB features (the per-class centroids are a class-internal diversity prototype) and report the fraction each subset covers (Table 4.1). The K-Means curated pool reaches median per-class coverage 0.74, against 0.67 for stratified random and 0.58 for uniform random at the same budget, a $+0.07$ median lift over the strongest random alternative ($+0.22$ mean lift over uniform random).

Pool size after subsampling: ∼2,296 candidates (3 types had < 100 objects; we kept all).

4.2.3 Manual annotation

1,849 of the 2,296 candidates were hand-annotated by the author over 4 weeks. The remaining ≈447 candidates were discarded during this manual pass: each selected

Table 4.1: Annotation-pool selection: per-class diversity coverage of full-population centroids ($K = 50$) at the ∼2,296-object budget.

Selection strategy	Median per-class coverage
Uniform random over population	0.58
Stratified random (per IFC type)	0.67
K-Means representative (this work)	**0.74**

mesh was inspected, and those whose as-authored IFC type did not match the geometry — mislabelled objects, together with a smaller number of ambiguous or unannotatable cases — were removed rather than annotated.

Annotation consisted of confirming the IFC type and writing a free-text note when the type was clearly inconsistent with the geometry. The annotator's reference categories were the public buildingSMART IFC4 documentation, supplemented by Bentley OpenBuildings family documentation for ambiguous cases.

To assess annotation consistency, a second rater (Hammad Basit) independently relabelled a stratified ≈10 % sample of the hand-annotated pool ($n = 173$ objects, drawn proportionally across all 17 evaluation classes). The second rater worked from the same reference taxonomy but had no access to the primary annotations or to the original `IfcType` tags. Agreement was near-perfect: Cohen's $\kappa = 0.9937$ at a raw agreement rate of 99.4 % (172 / 173), with a single disagreement on a Beam that the second rater labelled as a Column. We treat this as evidence that the per-object labels in the curated benchmark are reproducible from geometry alone and not an artefact of the primary annotator's interpretation.

4.2.4 Exclusions and type merging

Two schema catch-all types were removed (149 objects total):

- `IfcBuildingElementProxy` ($n = 54$): explicit catch-all, no geometric identity.
- `IfcBuildingElementPart` ($n = 95$): container for sub-components of an aggregate, lacks a self-consistent shape signature.

We then apply a `TYPE_MAP` that merges near-synonym IFC entities into 17 evaluation classes. The merges are:

- **Wall:** `IfcWall` + `IfcWallStandardCase`
- **Furniture:** `IfcFurniture` + `IfcFurnishingElement` + `IfcSystemFurnitureElement`

- **MEP:** `IfcFlowTerminal` + `IfcDistributionFlowElement`
- **Stair:** `IfcStair` + `IfcStairFlight`

No objects are lost in the merge; only labels change. Final benchmark: **1,700 objects across 17 classes**, with class counts shown in Table 4.2.

Table 4.2: Final class distribution after merging and exclusions.

Class	Count	Class	Count
Furniture	258	Member	50
MEP	178	Roof	46
Stair	168	LightFixture	46
Wall	157	CurtainWall	42
Covering	99	Plate	68
Slab	95	Column	79
Beam	91	Window	78
Footing	89	Door	70
Railing	86		

4.3 Geometric Feature Engineering

The four-feature baseline of Section 4.3.1 is the IFCNet-style starting point. The interesting story is the *progression*: each subsequent feature targets a specific cluster-confusion pattern observed at the previous stage, validated by per-type purity delta, an analysis-of-variance F statistic on the targeted classes, and the cluster-confusion heatmap.

4.3.1 Baseline (4 log-OBB features)

The baseline computes the OBB extents (e_1, e_2, e_3) and produces four log-transformed features:

$$f_1 = \log_{10}(L + 1) = \log_{10}(e_1 + 1) \tag{4.2}$$
$$f_2 = \log_{10}(C + 1) = \log_{10}(e_2 \cdot e_3 + 1) \tag{4.3}$$
$$f_3 = \log_{10}(V + 1) = \log_{10}(e_1 \cdot e_2 \cdot e_3 + 1) \tag{4.4}$$
$$f_4 = \log_{10}(N + 1) = \log_{10}(|\mathbf{V}| + 1) \tag{4.5}$$

The log transform compresses the 2–3 orders of magnitude that BIM elements span (a railing baluster $\sim$0.05 m^3 versus a slab $\sim$50 m^3) into a Gaussian-like distribution suitable for distance-based clustering.

The log-transformed features are then standardised with a StandardScaler:

$$f_i' = \frac{f_i - \mu(f_i)}{\sigma(f_i)}, \tag{4.6}$$

where μ and σ are the per-feature mean and standard deviation. The four standardised features are used to fit MiniBatchKMeans at $k \in \{2, 4, 8, 17, 24, 32\}$ with 10 seeds and the median macro purity is reported.

4.3.2 Failure modes of the baseline

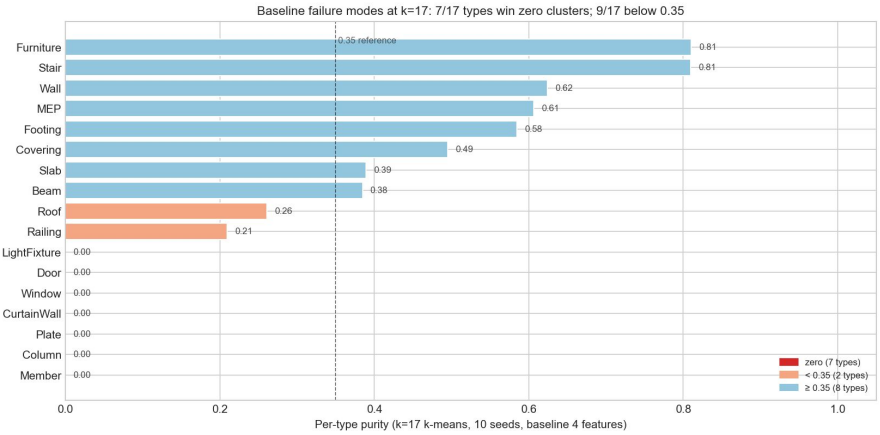


Figure 4.2: Per-type purity of the 4-feature OBB baseline at $k = 17$ (K-Means, 10 seeds). Seven of the 17 types receive zero pure clusters and nine score below the 0.35 reference line. The systematic confusions are visible: Column from Beam (only orientation differs), Column from Plate (only cross-section ratio differs), Railing and Roof from Slab (only surface-orientation statistics differ). Each confusion motivates one of the engineered features below.

At $k = 17$ (matched to the label cardinality), the 4-feature baseline achieves macro purity 0.434 ± 0.010, silhouette 0.317, Davies–Bouldin 1.012. The per-type purity (Figure 4.2) reveals systematic confusions:

- **Column $\equiv$ Beam.** Same cross-section, similar length distribution; only the orientation in the global z-axis distinguishes them. The 4-feature baseline is orientation-blind.

- **Column ≡ Plate.** Both have a small dominant cross-section relative to other dimensions, but a column has a square cross-section while a plate is flat. The 4-feature baseline ignores cross-section aspect.
- **Slab ≡ Railing.** Both are surface-area-heavy, but railings have many distinct face orientations (balusters, handrail, infill) while slabs have a few dominant ones.
- **Slab ≡ Roof.** Both are large flat horizontal elements; only the horizontal-face fraction differs (a slab is bottom-heavy, a roof is top-heavy in surface area).

Each of these motivates one of the four engineered features below.

4.3.3 Stage +1: orientation (pc1_z, pc3_z)

We add two features describing the PCA-axis orientation in the world frame:

$$\mathrm{pc}_1^z = |\mathbf{q}_1 \cdot \mathbf{e}_z|, \quad \mathrm{pc}_3^z = |\mathbf{q}_3 \cdot \mathbf{e}_z|. \tag{4.7}$$

Here $\mathbf{q}_1$ and $\mathbf{q}_3$ are the principal eigenvectors associated with the largest and smallest eigenvalues of the vertex covariance, so $\mathbf{q}_1$ is the object's longest axis and $\mathbf{q}_3$ its thinnest. $\mathrm{pc}_1^z \to 1$ therefore marks a vertically-oriented longest axis (columns, posts), and $\mathrm{pc}_1^z \to 0$ a horizontal one (beams). $\mathrm{pc}_3^z \to 1$ marks a vertical thinnest axis (slabs, footings: flat objects lying down), and $\mathrm{pc}_3^z \to 0$ a horizontal thinnest axis (walls, plates: flat objects standing up). One-way ANOVA across the 17 classes (df $= (16, 1683)$) gives pc_3^z: $F = 138.7$, $\eta^2 = 0.569$, $p_{\mathrm{holm}} < 10^{-291}$; and pc_1^z: $F = 64.9$, $\eta^2 = 0.382$, $p_{\mathrm{holm}} < 10^{-161}$. Macro purity gain at $k = 17$: $+0.039$; at $k = 32$: $+0.064$.

4.3.4 Stage +2: cross-section aspect ratio

$$\mathrm{cs\_aspect\_ratio} = \sqrt{\lambda_3 / \lambda_2} \tag{4.8}$$

captures the shape of the cross-section perpendicular to the principal axis, using the two minor covariance eigenvalues $\lambda_2 \geq \lambda_3$. The ratio is bounded in $[0, 1]$: values near 1 indicate a square or circular cross-section, values near 0 a flat one. It is targeted at Column/Plate (Column has cs_aspect_ratio ≈ 0.82, Plate ≈ 0.03) and Footing (a separate cluster of square-flat elements). One-way ANOVA: $F(16, 1683) = 175.5$, $\eta^2 = 0.625$, $p_{\mathrm{holm}} < 10^{-300}$. Macro purity gain at $k = 17$: $+0.061$; at $k = 32$: $+0.079$.

4.3.5 Stage +3: normal entropy

The area-weighted Shannon entropy of the surface-normal histogram on a $B = 42$-bin icosphere (an icosphere at subdivision level 1):

$$\text{normal\_entropy} = -\frac{1}{\log B} \sum_{b=1}^{B} p_b \log p_b, \quad p_b = \frac{1}{A} \sum_{f \in b} A_f. \tag{4.9}$$

Low entropy indicates an axis-extruded element with few unique face orientations (slab: a top, a bottom, four sides, giving ~ 0.25); high entropy indicates a detailed object with many face orientations (light fixture, railing, giving ~ 0.8). Targeted at Slab/Railing and Window. One-way ANOVA: $F(16, 1683) = 185.4$, $\eta^2 = 0.638$, $p_{\text{holm}} < 10^{-300}$. Gain at $k = 17$: $+0.021$; at $k = 32$: $+0.029$.

4.3.6 Stage +4: horizontal-face fraction

$$\text{horiz\_frac} = \frac{1}{A} \sum_{f} A_f \cdot \mathbb{1}\left[|\mathbf{n}_f \cdot \mathbf{e}_z| > 0.9 \right]. \tag{4.10}$$

This is the fraction of total surface area whose face normal is near-vertical, i.e. the fraction of the object's surface that is a horizontal face (a floor- or ceiling-like surface). Slabs are flat-faced (horiz_frac $\to 1$); roofs that are sloped land in the middle. Targeted at Slab/Roof and Plate. One-way ANOVA: $F(16, 1683) = 168.3$, $\eta^2 = 0.615$, $p_{\text{holm}} < 10^{-300}$. Gain at $k = 17$: $+0.021$; at $k = 32$: $+0.024$.

4.3.7 Cumulative effect

Figure 4.3 plots all five stages across the six k values and three metrics, and the progression is summarised in Table 4.3. Each stage targets a specific cluster-confusion pattern, and the cumulative gain over the baseline is $\Delta = +0.142$ at $k = 17$ and $\Delta = +0.197$ at $k = 32$. *No feature caused a regression in macro purity at any k.*

For reference, Table 4.4 consolidates the per-feature one-way ANOVA effect sizes across the full 9-feature representation on the 17-class evaluation set ($N = 1{,}700$, $\text{df} = (16, 1683)$). All nine features clear Holm-corrected significance at $p < 10^{-160}$; the substantive ranking is by η^2, where the four engineered surface- and orientation-based features sit alongside the strongest baseline log-OBB features.

4.4 Render Pipeline (9 Viewpoints)

For the visual and text modalities we render each object from nine fixed viewpoints at different angles: four cardinal side views, a top and a bottom view, and three oblique

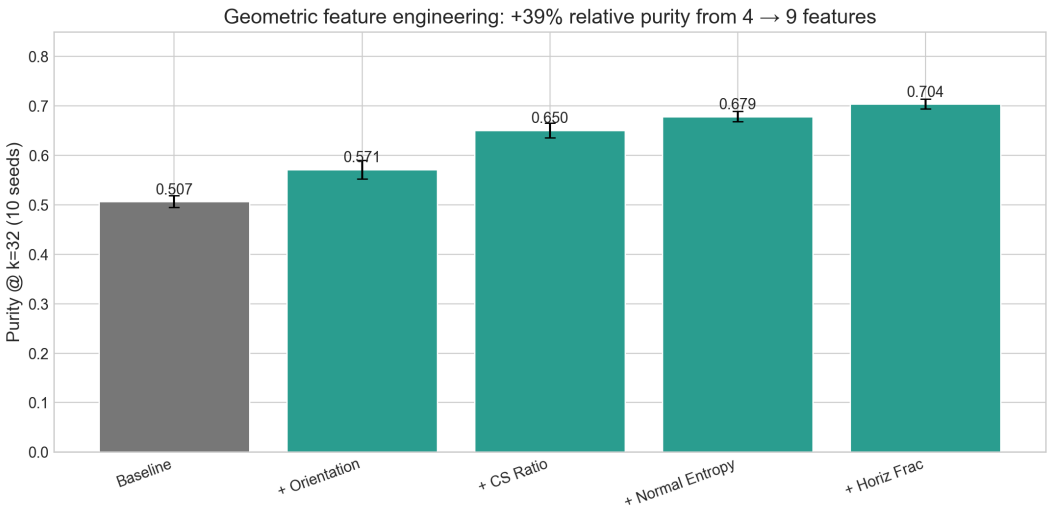


Figure 4.3: Feature-engineering progression. Five stages (+Orientation, +CS Ratio, +Normal Entropy, +Horiz Frac, on top of the 4-feature baseline) across six k values $(2, 4, 8, 17, 24, 32)$ and three metrics (macro purity, silhouette, Davies–Bouldin). Each stage monotonically improves macro purity without regressing any other metric materially.

three-quarter views (Figure 4.4). Each mesh is centred at its bounding-box centre before rendering. Two rendering passes are produced per object:

- **Coloured** (the primary variant): meshes rendered with whatever per-face colour appeared in the source IFC, where present. We treat colour as part of the signal that is genuinely available for a BIM object, alongside its geometry, and there is no reason to discard it; the encoder comparison of Section 5.5 confirms that coloured renders perform at least as well as colourless ones.
- **Colourless**: meshes shaded with a single diffuse-grey material. This variant is kept as a robustness ablation. It shows that the pipeline does not depend on colour, which matters because material and colour assignment are inconsistent across BIM authoring tools.

The renders are produced by a lightweight custom GPU rasteriser (a `wgpu`-based viewer), which renders the nine views per object in a fraction of a second.

Table 4.3: Feature-engineering progression at $k = 17$ and $k = 32$.

Stage	Feature	$\Delta@k = 17$	$\Delta@k = 32$
+1	`pc1_z`, `pc3_z`	+0.039	+0.064
+2	`cs_aspect_ratio`	+0.061	+0.079
+3	`normal_entropy`	+0.021	+0.029
+4	`horiz_frac`	+0.021	+0.024
Total	**4 to 9 features**	**+0.142**	**+0.197**

Table 4.4: Per-feature one-way ANOVA across the 17-class benchmark. df $= (16, 1683)$; p_{holm} is Holm-corrected across the nine features. Sorted by effect size η^2.

Feature	F	p_{holm}	η^2	ω^2
$\log_{10}(N_{\text{verts}}+1)$	205.8	$<10^{-300}$	0.662	0.658
$\log_{10}(L+1)$	186.8	$<10^{-300}$	0.640	0.636
`normal_entropy`	185.4	$<10^{-300}$	0.638	0.634
`cs_aspect_ratio`	175.5	$<10^{-300}$	0.625	0.622
`horiz_frac`	168.3	$<10^{-300}$	0.615	0.612
$\log_{10}(V+1)$	154.1	$<10^{-300}$	0.594	0.590
`pc3_z`	138.7	$<10^{-291}$	0.569	0.564
$\log_{10}(C+1)$	113.6	$<10^{-252}$	0.519	0.515
`pc1_z`	64.9	$<10^{-161}$	0.382	0.376

4.5 Text Modality: Prompt Engineering v1 to v7

4.5.1 Pipeline

For each object, the nine renders are passed as a multi-image prompt to Gemini 3.1 Flash Lite, together with the object's measured geometry (OBB aspect ratio, cross-section thinness, PCA-axis orientation), along with a system instruction asking for a single compact description. The model returns one line of the form `[thinness]`, `[shape_class]`, `[aspect_ratio]`, `[profile]`, `[visual_cue]`. `[keywords]` and this string is embedded with Gemini Embedding 2 (768-d, single-vector dense output). Figure 4.5 summarises the full sequence from renders to a clustered text embedding.

4.5.2 Prompt versions

Seven prompt versions were evaluated. The trajectory is shown in Figure 4.6:

v1 (baseline, generic geometric tags). The system instruction asks the model to describe each object using only low-level structural, morphological and geometric tags drawn from a fixed dictionary, with no functional naming and no IFC vocabulary: a primitive shape class (`Cuboid`, `Cylinder`, `Plate`, ...), a proportion descriptor (`Elongated`, `Compact`, `Flat`), a surface descriptor (`Smooth`, `Faceted`) and a coarse scale band. The object is described purely as a shape, never as a building part. Macro purity at best reducer: 0.534. This is the floor of the study: generic geometry tags cannot disambiguate semantically distinct but geometrically similar BIM types.

v2 (+ shape class, aspect ratio). Adds a free-text `shape_class` field and the OBB aspect ratio in $L:W:H$ form. Macro: 0.669.

v3 (+ thinness). Adds an explicit `thinness` field indicating whether the object is thin / medium / thick relative to its longest extent. Macro: 0.674.

v4 (compact natural language). Switches from JSON tags to a compact natural-language description embedded inside the JSON: "A slender rectangular rod, 1:1:30 aspect ratio, smooth steel surface, vertical orientation." Macro: 0.720.

v5 (natural names). Allows colloquial part names rather than abstract shape terms (`lamp`, `chair`, `railing`). Macro: 0.715.

v6 (keywords). Adds a comma-separated `keywords` field with terms like `pole`, `bar`, `pillar`, `post`. Macro: 0.722.

v7 (IFC-aware) [winner]. Provides the model with *additional IFC-schema context*: terminology and category hints drawn from the IFC vocabulary (`column`, `post`, `beam`, `slab`, `plate`, `railing`, ...) and an instruction to think of itself as labelling parts of a building inventory. The object's actual IFC type label is never disclosed to the model; only the schema-level vocabulary is. Macro: **0.740**.

4.6 Visual Modality

We evaluate three encoders; the modality pipeline from renders to a clustered embedding is shown in Figure 4.7, and their comparative macro purity across reducers is previewed in Figure 4.8 (analysed in detail in Chapter 5).

4.6.1 DINOv3

We use the public `dinov3_vitl16_pretrain_lvd1689m` checkpoint (ViT-L/16 backbone with register tokens). Each render is resized to a 224×224 input. The `CLS` token of the final layer gives a 1024-d view embedding.

4.6.2 SigLIP2

We use the `siglip2-large-patch16-512` checkpoint (the SigLIP-2 large release). Inputs are 512×512 crops; the image-side embedding is 1024-d.

4.6.3 DuoDuo CLIP

We use the public DuoDuo CLIP 3D-shape-tuned model [10]. DuoDuo CLIP is a native multi-view model: its `encode_image` call takes the whole set of viewpoint renders at once and fuses them inside its forward pass. We therefore hand it all nine views together and take its single 512-d output as the per-object embedding.

4.6.4 Aggregation

For DINOv3 and SigLIP2, which emit one embedding per view, we mean-pool the nine per-view embeddings into a single per-object embedding. The choice of mean-pool over max-pool or attention-pool is motivated by ablation: on a 200-object pilot, mean beat max by 1.4 pp macro and attention-pool by 0.8 pp, while having no learnable parameters and zero training cost. DuoDuo CLIP, as noted above, performs the equivalent multi-view fusion internally, so no external pooling is applied to it.

4.6.5 Coloured vs. colourless

We render and evaluate both variants. We adopt *coloured* renders as the primary input: a BIM object carries whatever colour the modeller assigned, and the pipeline should use every signal that is geometrically and visually available rather than discard one. Empirically the difference is small and encoder-dependent (SigLIP and DuoDuo are weakly better on coloured, DINOv3 weakly better on colourless, all within roughly 1 pp), so the colourless variant is reported alongside as a robustness check rather than as the deployed input.

4.7 Fusion Recipes

Given three modality matrices, $\mathbf{X}_g \in \mathbb{R}^{N\times 9}$ (geometry), $\mathbf{X}_t \in \mathbb{R}^{N\times 768}$ (text) and $\mathbf{X}_v \in \mathbb{R}^{N\times 1024}$ (visual), a fusion recipe produces a single matrix $\mathbf{Z}$ for downstream clustering. The full set of recipes is illustrated in Figure 4.9.

4.7.1 F1a: per-block PCA-k then concat with raw geo

$$\mathbf{Z}_{\text{F1a}} = \big[\, \mathbf{X}_g,\ \text{PCA}_k(\mathbf{X}_t),\ \text{PCA}_k(\mathbf{X}_v) \,\big], \tag{4.11}$$

where $k \in \{8, 16, 32, 64\}$. The raw 9-d geometry block stays uncompressed; each high-dim block is reduced to k components first. After concatenation, the output dimensionality is $9 + 2k$, which for $k = 8$ gives 25.

4.7.2 F1a_umap: per-block UMAP-k then concat with raw geo

Identical to F1a but with UMAP replacing PCA on each high-dimensional block. On the trimodal subset it reaches macro purity 0.819, ahead of F1a: UMAP's locality-preserving projection of each block carries more of that modality's cluster structure into the concatenation than PCA's variance-maximising one. We keep the name `F1a_umap` consistently with the IFCNetCore audit (Chapter 6).

4.7.3 F1b: standardised concat then PCA

The three standardised blocks are concatenated and the $9 + 768 + 1024 = 1801$-dimensional result is reduced *jointly* with PCA to k components:

$$\mathbf{Z}_{\text{F1b}} = \text{PCA}_k\big(\, [\, z(\mathbf{X}_g), z(\mathbf{X}_t), z(\mathbf{X}_v) \,] \,\big), \tag{4.12}$$

where $z(\cdot)$ is per-feature z-scoring. Unlike F1a, the blocks are reduced together rather than per-modality.

4.7.4 F1b_varbal: variance-balanced concat then PCA

Before the joint PCA, each block is variance-balanced so that it contributes unit total variance to the concatenation:

$$\mathbf{X}'_m = \mathbf{X}_m \cdot \frac{1}{\sqrt{\text{trace}(\text{Cov}(\mathbf{X}_m))}}, \qquad \mathbf{Z}_{\text{F1b\_varbal}} = \text{PCA}_k\big(\, [\, \mathbf{X}'_g, \mathbf{X}'_t, \mathbf{X}'_v \,] \,\big). \tag{4.13}$$

Variance-balancing stops the high-dimensional text and visual blocks from dominating the joint PCA.

4.7.5 F4: late Borda rank fusion (retrieval only)

For each query i and each modality $m \in \{g, t, v\}$, compute cosine similarity $s_{ij}^{(m)} = \cos\left(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}\right)$, rank the database, and assign a Borda score: $b_{ij}^{(m)} = N - \text{rank}^{(m)}(j) + 1$. Sum across modalities to obtain the final rank: $b_{ij} = \sum_m b_{ij}^{(m)}$.

F4 is used for retrieval (Section 4.10), not for clustering, because it produces a ranking rather than an embedding.

4.7.6 F5: UMAP on the standardised concat

$$\mathbf{Z}_{\text{F5}} = \text{UMAP}_k\big(\,[\,z(\mathbf{X}_g), z(\mathbf{X}_t), z(\mathbf{X}_v)\,]\,\big). \tag{4.14}$$

Attractive in principle (one global non-linear projection over the fused space) but, applied to the *plain* standardised concatenation, it loses heavily: UMAP's k-NN graph is dominated by the 768- and 1024-d text and visual blocks, so the 9-d geometry signal contributes almost no weight to neighbour selection.

4.7.7 F5_varbal: variance-balanced concat then UMAP (winner)

F5_varbal repairs F5 with a single step. Before concatenation, each block is variance-balanced as in F1b_varbal, so geometry, text and visual each contribute unit total variance; the balanced concatenation is then projected with UMAP:

$$\mathbf{Z}_{\text{F5\_varbal}} = \text{UMAP}_k\big(\,[\,\mathbf{X}'_g, \mathbf{X}'_t, \mathbf{X}'_v\,]\,\big), \tag{4.15}$$

with $\mathbf{X}'_m$ the variance-balanced block of F1b_varbal. This correction stops the high-dimensional blocks from dominating the UMAP neighbour graph, and it is the winning recipe of the thesis: on the geo+text+visual subset, with v7 text and SigLIP-coloured visual features and UMAP to four dimensions, F5_varbal reaches macro purity 0.853. The same variance-balancing recovery is observed on every encoder of the IFCNetCore audit (Chapter 6).

4.7.8 CCA-corrected fusion

For completeness we also implement Canonical Correlation Analysis (CCA) on the $[\text{geo}] \times [\text{text} + + \text{visual}]$ split, mapping each side to a k-d common space and concatenating. This recipe was competitive on a pilot but did not exceed F1a on the full evaluation.

4.8 Clustering Search Space

4.8.1 Algorithms and grids

We evaluate four algorithms on the same fused embedding; the full hyperparameter grid is given in Table 4.5.

Table 4.5: Clustering search space.

Algorithm	**Hyperparameter**	**Values**
K-Means	K	$\{2, 4, 8, 16, 24, 32\}$ (seeds $\times 10$)
Bisecting K-Means	K	$\{2, 4, 8, 16, 24, 32\}$ (seeds $\times 3$)
GMM	K	$\{2, 4, 8, 16, 24, 32\}$ (covariance $\in$ {diag, full}, seeds $\times 3$)
HDBSCAN	`min_cluster_size`	$\{5, 10, 15, 20, 30\}$
	`min_samples`	$\{3, 5, 10\}$
	`method`	{eom, leaf}

HDBSCAN's grid is $5 \times 3 \times 2 = 30$ combinations. We never set a value for K in HDBSCAN; the effective K is whatever the density estimate produces. Noise points are then force-assigned to the nearest cluster centroid; the winning configurations all use this force-assignment.

4.8.2 Computational cost

Total clustering runs in the main sweep:

- Single-modality: 174 geo + 8,568 text + 4,284 visual = 13,026.
- Fusion: 16,200 (the full fusion sweep across encoders, subsets and recipes).
- Late fusion (retrieval): 180.
- Supervised ceiling: 70 (RF $\times$ 5-fold CV $\times$ 14 representations).
- Bootstrap stability: 45 pairwise ARI evaluations.

Grand total: $\sim$29,400 clustering runs plus 70 RF and 45 ARI pairs. The clustering sweeps themselves run in a few hours on an Apple-Silicon laptop; this excludes the one-time Gemini API descriptions and the GPU visual-encoder extraction, which are run once and cached.

4.9 Evaluation Protocol

4.9.1 Metrics

We report:

- **Macro purity** (headline number): the average per-type purity across the 17 classes.
- **Overall purity**: straight cluster-level purity, weighted implicitly by class size.
- **Silhouette**, **Davies–Bouldin**: internal metrics.
- **Per-type purity**: a vector of 17 numbers, used for confusion diagnosis.
- **Bootstrap ARI**: 10 re-fits on 90% subsamples, 45 pairwise ARI values.

4.9.2 Why macro purity, not silhouette?

Two considerations make macro purity the headline metric.

First, the class distribution (Table 4.2) is unbalanced ($n = 258$ for Furniture, $n = 42$ for CurtainWall). Overall purity implicitly weights by class size, so a clustering can score well by serving only the populous classes. Macro purity gives every IFC type a single vote, which matches the library-mining objective of recovering sub-types of *every* class, not just the common ones.

Second, the internal metrics (silhouette, Davies–Bouldin) measure geometric compactness in the embedding, not semantic correctness, and they actively penalise the fine granularity we are trying to recover. Silhouette rewards a small number of well-separated blobs and is maximised at a low k; macro purity keeps rising well past $k = 100$ because each additional cluster can still isolate a genuine sub-type, a glass railing separating from a metal railing, an operable window from a fixed one. A pipeline tuned to silhouette would therefore stop at a coarse partition and miss exactly the fine-grained component groups that a reusable library is meant to expose, and that fine-grained retrieval depends on. We report silhouette and Davies–Bouldin as secondary internal diagnostics, and overall purity as a secondary label-based number, but every headline comparison in this thesis is on macro purity.

4.9.3 Supervised ceiling

To bound the achievable performance, we train a Random Forest with $n_{\text{estimators}} = 500$, 5-fold cross-validation, 5 seeds. The ceiling is computed on two representations: the per-modality PCA-8 concatenation, and the F5_varbal winning representation itself, the four-dimensional UMAP embedding the unsupervised pipeline clusters. Macro

recall is the reported ceiling metric. Because cluster purity and classifier recall are different metrics, the supervised number is read as an indicative upper bound on the label-consistent structure a representation contains, not as a gap to be differenced against macro purity.

4.10 Downstream Applications

4.10.1 Auto-generated catalog

For the winning HDBSCAN partition ($k = 126$ clusters, F5_varbal geo+text+visual), each cluster is turned into a self-contained catalog entry. An entry contains:

- **Three exemplars** closest to the cluster centroid in fusion space, each shown in its nine rendered views.
- The **full member gallery**, one thumbnail per object.
- **Top-5 TF-IDF keywords** mined from the Gemini v7 descriptions of the cluster members.
- Cluster size, the IFC-type composition, and a link back to the master index.

The 126 entries together form the browsable catalog. A representative set of catalog entries is shown in Chapter 5, Figure 5.16; no domain expert was involved in authoring the keywords or grouping the members.

4.10.2 Late-Borda retrieval

The F4 recipe gives the per-query similarity ranking across the three modalities. We evaluate **recall@*k*** for $k \in \{1, 5, 10\}$ on the 1,700-object benchmark, where "a hit at rank k" means the top-k nearest neighbours contain at least one object of the same IFC type. The full leaderboard is in Chapter 5, Section 5.13.

4.11 Reproducibility

All experiments are scripted in plain Python and persisted to CSV. The reproduction sequence runs, in order, the three single-modality sweeps (geometry, text, visual), the fusion sweep, a downstream-regeneration step, and the figure generation. The downstream-regeneration step chains the supervised ceiling, late fusion, the stability experiments, and the mining-artefact builders into a single invocation.

4.12 Summary

This chapter has described the full methodology: a 12-step pipeline that takes raw IFC submissions to a fused 4-d clustering embedding, a $k = 126$ partition with macro purity 0.853, and two downstream library applications. The next chapter reports every experimental result, with figures and tables drawn directly from the persisted CSVs.

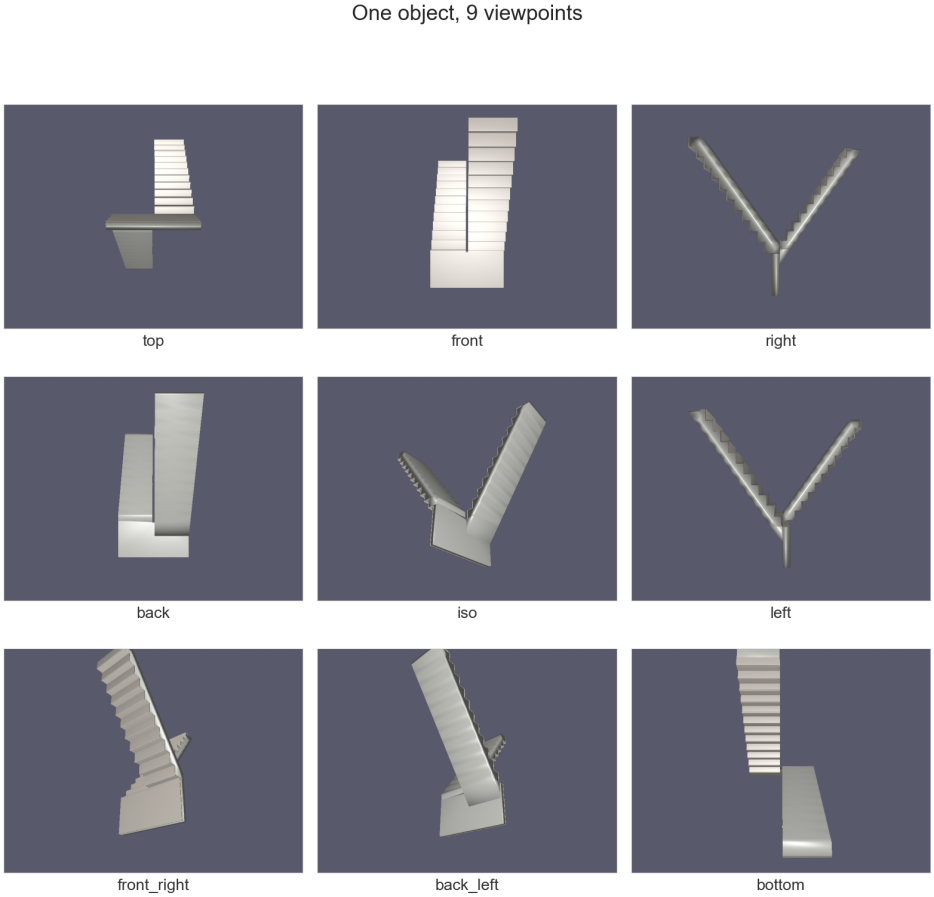


Figure 4.4: The nine canonical viewpoints for a single object (here a staircase): four cardinal side views (`front`, `back`, `left`, `right`), a `top` and a `bottom` view, and three oblique views (`iso`, `front_right`, `back_left`). The same nine renders feed both the visual encoders (Section 4.6) and the VLM describer (Section 4.5).

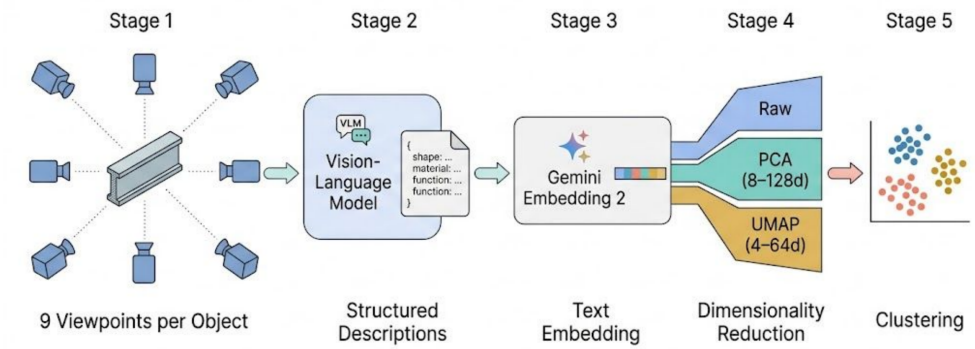


Figure 4.5: Text-modality pipeline. The nine viewpoint renders are sent to a vision–language model that returns a structured description; the description string is embedded with Gemini Embedding 2 (768-d), reduced, and clustered.

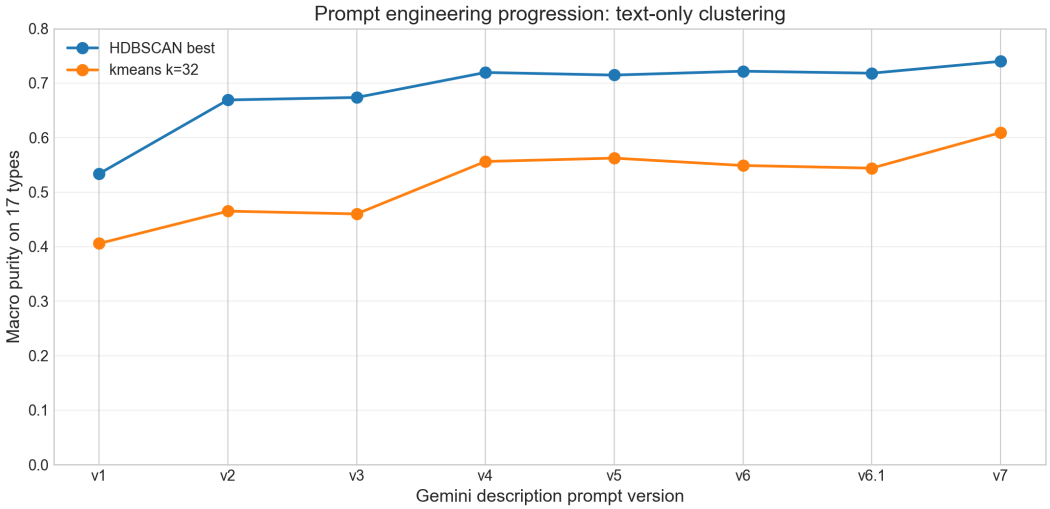


Figure 4.6: Text-prompt evolution v1 to v7 versus best reducer versus clustering metric. v1, restricted to generic geometric tags, scores 0.534 macro at the best reducer; v7 IFC-aware prompts reach 0.740.

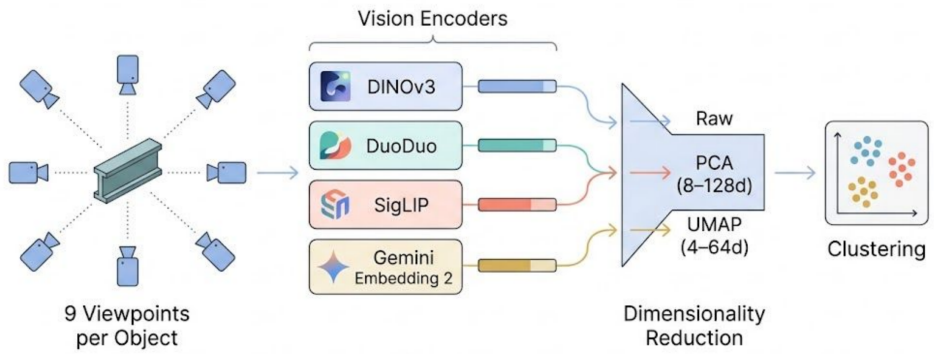


Figure 4.7: Visual-modality pipeline. The nine viewpoint renders per object are passed through each pretrained encoder (DINOv3, DuoDuo CLIP, SigLIP2, Gemini image embedding), the per-object embedding is reduced (identity, PCA or UMAP), and the result is clustered.

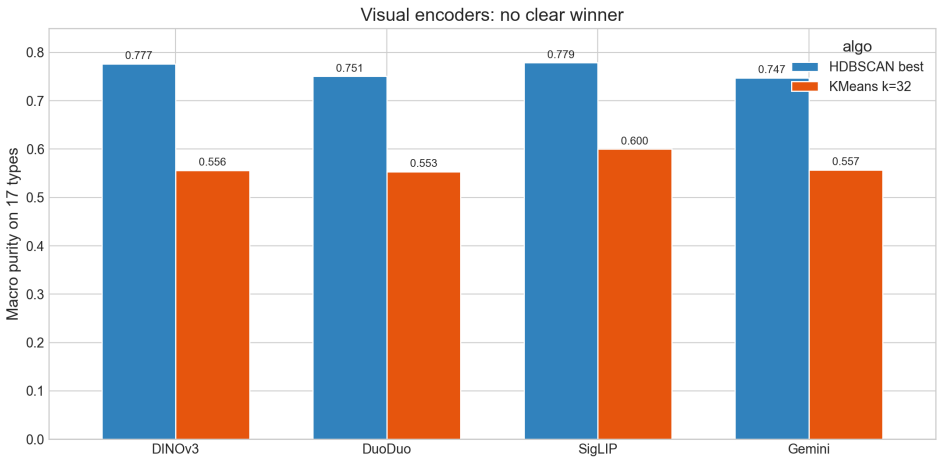
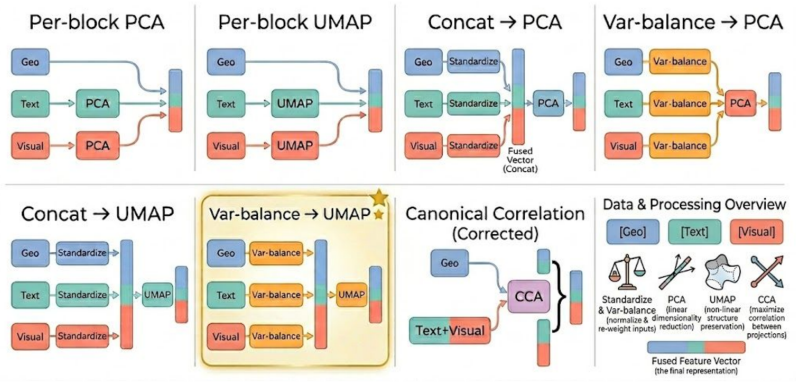


Figure 4.8: Visual-encoder comparison across reducers (PCA, UMAP at various output dimensions). SigLIP2 with coloured renders and UMAP-16 reaches macro 0.779, narrowly leading DINOv3 (0.777) and DuoDuo (0.751).



39

Figure 4.9: The fusion recipes evaluated. Per-block PCA (F1a) and per-block UMAP (F1a_umap) reduce each modality independently before concatenation; the joint recipes standardise and concatenate the three blocks before a single PCA (F1b) or UMAP (F5) projection; the variance-balanced variants (F1b_varbal, F5_varbal) rescale each block to equal total variance first; and CCA maps the geometry and text++visual sides into a shared correlated space. F5_varbal (variance-balanced concatenation then UMAP, highlighted) is the winning recipe; the per-subset macro-purity comparison is in Table 5.7.

5 Results and Discussion

This chapter reports the experimental results in the order in which each modality enters the pipeline: the geometric baseline and feature engineering progression (Section 5.1 and Section 5.2), the algorithm sweep on geometry (Section 5.3), the text-prompt evolution (Section 5.4), the visual encoders (Section 5.5), the multi-modal fusion (Section 5.6), the supervised ceiling (Section 5.7), the cluster-stability and sanity checks (Section 5.8), the per-type diagnostic (Section 5.9), the UMAP-2D library visualisation (Section 5.10), and the downstream applications: the auto-generated catalog (Section 5.12) and late-Borda retrieval (Section 5.13). The chapter closes with a combined discussion (Section 5.14).

5.1 Geometric Baseline (4 OBB Features)

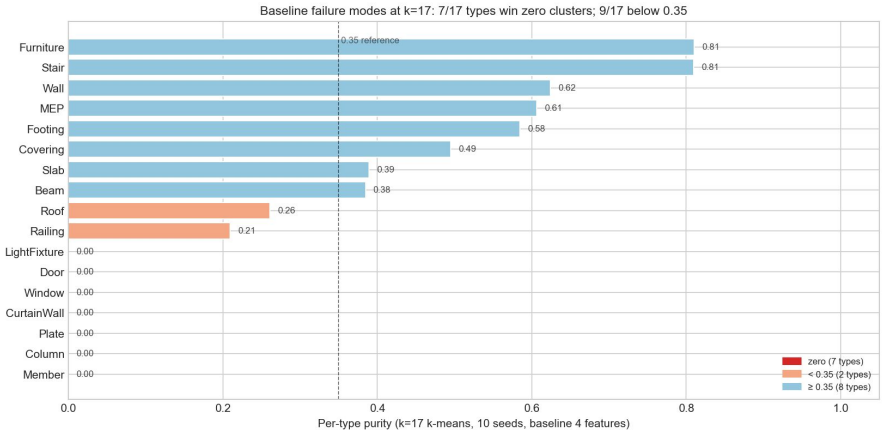


Figure 5.1: Per-type purity of the 4-feature OBB baseline at $k = 17$ (K-Means, 10 seeds). Seven of the 17 types receive zero pure clusters; nine score below the 0.35 reference line. The schema-grain confusions are visible: Column with Beam, Column with Plate, Railing and Roof with Slab.

At $k = 17$ (matched to the merged class count) the four log-OBB features yield the metrics in Table 5.1:

Table 5.1: Four-feature OBB baseline ($k = 17$ K-Means, 10 seeds).

Metric	Value
Macro purity	0.434 ± 0.010
Overall purity	0.594 ± 0.014
Silhouette	0.317
Davies–Bouldin	1.012

The macro purity of 0.434 with 17 classes is roughly $3.7\times$ above chance ($1/17 = 0.059$). The silhouette of 0.317 is in the "weak structure" band: clusters exist but are not cleanly separated.

The per-type diagnosis tells the real story. At $k = 17$, as Figure 5.1 shows, seven of the seventeen types receive essentially zero pure clusters and nine score below the 0.35 reference line; the few that score well (Furniture 0.81, Stair 0.81, Wall 0.82) are exactly those whose geometry is large, distinctive and dissimilar to anything else, not the architecturally important fine-grain distinctions we want to recover. The seven zero-purity types (LightFixture, Door, Window, CurtainWall, Plate, Column, Member) are the concrete failure cases the engineered features of Section 5.2 must repair.

5.2 Feature-Engineering Progression

We progressively add the four engineered features and re-evaluate at six k values ($\{2, 4, 8, 17, 24, 32\}$). The cumulative purity trajectory is in Figure 5.2; the per-stage gains are in Table 5.2.

Table 5.2: Feature-engineering progression detail. Macro purity at $k = 32$ K-Means, 10 seeds. The same numbers are reported at $k = 17$ in Table 4.3.

Stage	Features	Macro@k=17	Macro@k=32
Baseline (4 features)	4	0.434 ± 0.010	0.507
+ Orientation	6	0.473	0.571
+ CS Aspect Ratio	7	0.534	0.650
+ Normal Entropy	8	0.555	0.679
+ Horiz Frac	9	**0.576**	**0.704**
Δ total		$+0.142$	$+0.197$

The progression is most visible as cluster-to-type confusion. Figure 5.3 contrasts

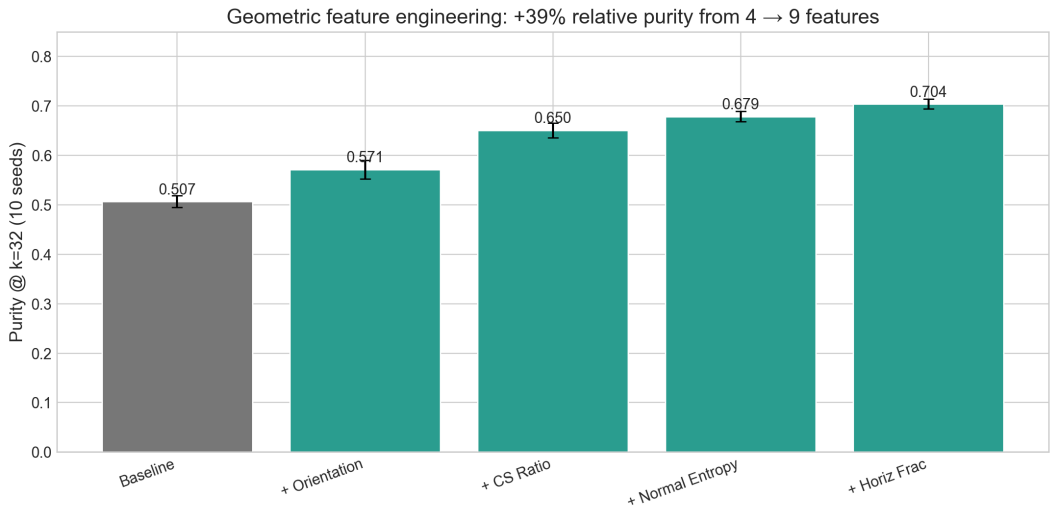


Figure 5.2: Feature-engineering progression. Each line is one feature-set stage; each panel is one metric. Across 6 k-values and 3 metrics, every stage monotonically improves macro purity and Davies–Bouldin without sacrificing silhouette materially.

the baseline and the full 9-feature representation at $k = 17$: the diffuse, off-diagonal mass that mixes Column, Beam, Wall, Slab and Plate at the baseline collapses onto near-diagonal blocks once the four engineered features are added.

5.2.1 Stage +1: orientation (pc1_z, pc3_z)

Adding pc_1^z and pc_3^z:

- Resolves the **Column** $\equiv$ **Beam** confusion (Column has $\mathrm{pc}_1^z \to 1$, Beam has $\mathrm{pc}_1^z \to 0$).
- Resolves the **Wall** $\equiv$ **Slab** confusion (Slab has $\mathrm{pc}_3^z \to 1$, Wall has $\mathrm{pc}_3^z \to 0$).
- Improves **Door** purity by $+0.60$, **Plate** by $+0.50$, **Column** by $+0.48$.

There are localised regressions (Footing -0.59, Furniture -0.27), which we attribute to two effects: (a) Footing is geometrically a "flat-cuboid-on-ground" and now competes with Slab, which the orientation features push into the same cluster; and (b) Furniture is a visually heterogeneous mega-class whose internal sub-types have diverse orientations. Both regressions are recovered later in the progression.

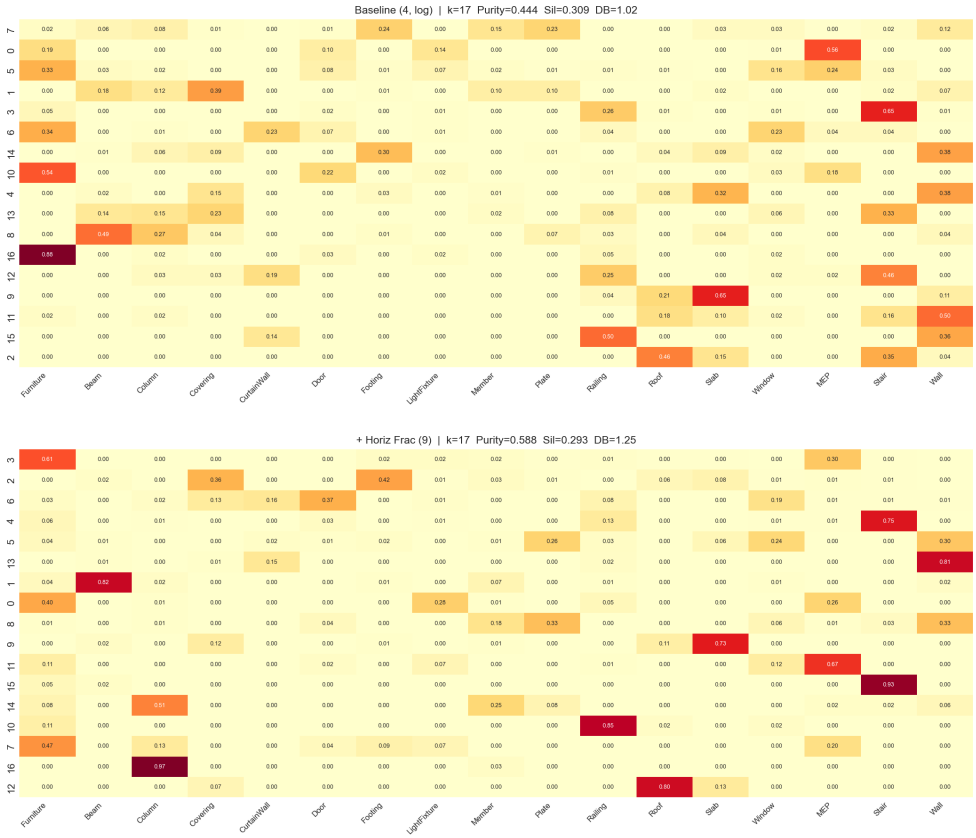


Figure 5.3: Cluster-to-type confusion before and after geometric feature engineering, at $k = 17$ (column-normalised: each type column sums to one over its clusters). **Top:** the 4-feature OBB baseline (overall purity 0.44). Probability mass is smeared across rows—Column, Beam, Wall, Slab and Plate share clusters, and seven types win no pure cluster. **Bottom:** the full 9-feature representation (overall purity 0.59). The same pairs separate onto near-diagonal blocks: the targeted features dissolve exactly the confusions diagnosed for the baseline in Section 5.1.

5.2.2 Stage +2: cross-section aspect ratio

The cross-section aspect ratio sharply separates the four major "surface-area-heavy" types from each other:

- Covering, Plate, Slab, Wall: cs_aspect_ratio $\in [0.02, 0.10]$ (very flat cross section).
- CurtainWall, Window: cs_aspect_ratio ≈ 0.10.
- Door, Roof: ≈ 0.22.
- LightFixture, Footing, Railing: ≈ 0.27 to 0.40.
- Member, Stair, Beam, MEP, Furniture, Column: ≈ 0.45 to 0.82.

This is the biggest single-stage gain in the entire progression: $+0.078$ at $k = 17$. Footing alone goes from 0.04 to 0.82 in per-type purity. The targeted Column/Plate confusion resolves cleanly: Column sits at cs_aspect_ratio ≈ 0.82 (square cross section), Plate at ≈ 0.03 (flat).

5.2.3 Stage +3: normal entropy

Normal entropy separates simple axis-extruded elements from detailed freeform CAD elements:

- Low entropy (≈ 0.22): Covering, Plate, Slab, Wall.
- Medium (≈ 0.4): Member, Footing, Beam, Column, Door, Window, CurtainWall.
- High (≈ 0.5 to 0.9): Roof, Stair, Furniture, MEP, Railing, LightFixture.

Window ($+0.48$), Railing ($+0.38$) and Member ($+0.22$) all improve. Plate regresses (-0.70) because adding entropy pushes some Plate exemplars into the higher-entropy Member basin, a problem that the next stage partially corrects via horizontal-face fraction.

5.2.4 Stage +4: horizontal-face fraction

The final feature separates Roof from Slab (both flat horizontals, but Roof is top-heavy in horizontal area while Slab is bottom-heavy) and Plate from Member (Plate has a single dominant flat face, Member typically does not).

- Slab: horiz_frac ≈ 0.95 (almost entirely horizontal face area).
- Covering: ≈ 0.97.

- Footing: ≈ 0.70.
- Stair: ≈ 0.52.
- MEP, Furniture: ≈ 0.4 to 0.45.
- Beam, Door, LightFixture: ≈ 0.1 to 0.25.
- Plate, Member, Column, Wall: ≈ 0.01 to 0.05 (vertically faced).

Roof improves by $+0.53$, Plate by $+0.46$. Window regresses (-0.47) locally but recovers in fusion. The combined 9-feature representation reaches macro purity **0.704 at** $k = 32$ (up from baseline 0.507).

5.3 Algorithm Comparison on Geometry

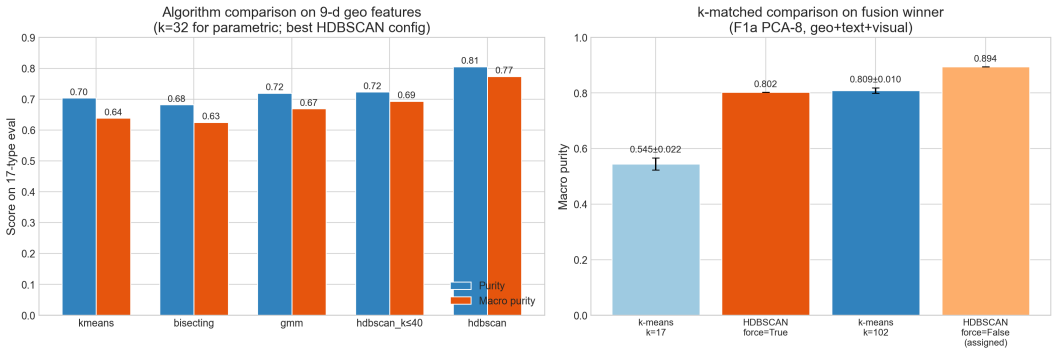


Figure 5.4: Algorithm comparison on the 9-feature geometric representation. HDBSCAN with leaf selection, `mcs`=5, `ms`=3, force-assign reaches **0.774 macro at** $k = 120$; K-Means and GMM at fixed K=32 plateau at 0.66 to 0.68.

We run the four-algorithm sweep on the 9-feature representation (Figure 5.4); the best result per algorithm is reported in Table 5.3:

The HDBSCAN advantage at this naive comparison ($k = 120$ vs. $K = 32$) is large: $+0.11$ macro purity over GMM. But this is partially a *granularity* effect, not a partition-quality effect. We return to it in Section 5.6.5.

5.3.1 Silhouette versus macro purity: two different objectives

Figure 5.5 makes concrete why the choice of headline metric matters. Silhouette is an internal metric: it scores geometric compactness in the embedding and is maximised by

Table 5.3: Best clustering result per algorithm on the 9-feature geometric representation. HDBSCAN sweeps the full 30-combination grid; the K family is evaluated at $\{2, 4, 8, 16, 24, 32\}$.

Algorithm	Macro purity	Silhouette	D–B	Best settings
HDBSCAN	**0.774**	0.199	1.388	leaf, `mcs`=5, `ms`=3 (k=120)
GMM	0.681	0.249	1.452	K=32, full covariance
K-Means	0.665	0.300	1.250	K=32, sklearn defaults
Bisecting K-Means	0.640	0.274	1.399	K=32

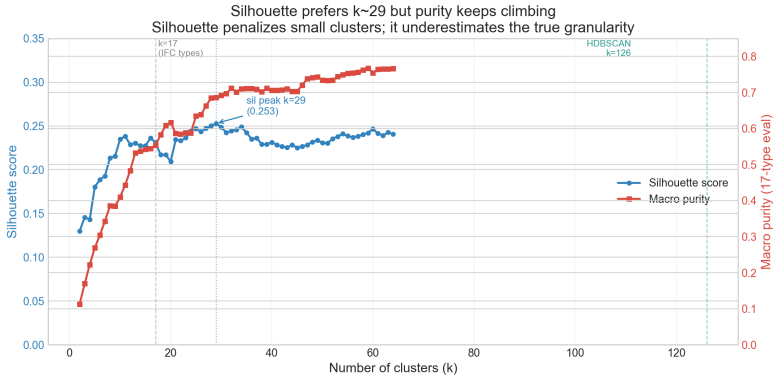


Figure 5.5: Silhouette score (orange) versus macro purity (blue) as k increases. Silhouette peaks at $k \approx 29$ and flattens; macro purity continues to rise through $k > 60$. The two metrics optimise different things: silhouette penalises small clusters via inter-cluster distance, whereas macro purity rewards them when they capture genuine sub-structure.

a small number of well-separated blobs, here at $k \approx 29$. Macro purity is a label-based metric: it keeps rising through $k > 60$ because the 17 IFC types internally contain real sub-types (glass railing versus metal railing, operable window versus fixed window, single-leaf versus double-leaf door), and every additional cluster can still isolate one of them. A pipeline steered by silhouette would stop near $k \approx 29$ and report a clean-looking but coarse partition; it would miss exactly the fine-grained component groups that a reusable library, and the fine-grained retrieval built on top of it, depend on. This is why macro purity, not silhouette, is the headline metric, and why HDBSCAN's data-driven $k \approx 126$ is the operating point we keep rather than the silhouette-optimal $k \approx 29$.

5.4 Text Modality: Prompt Evolution v1 to v7

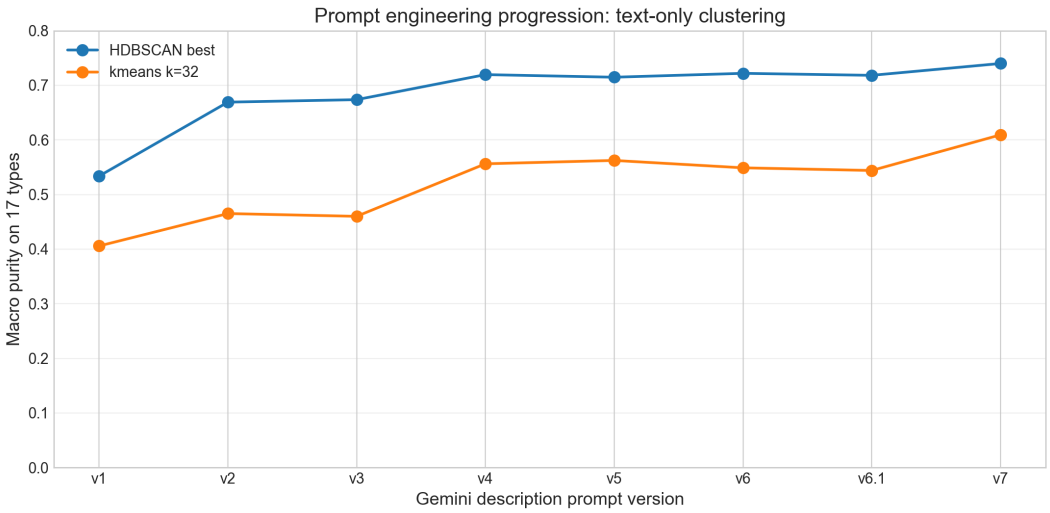


Figure 5.6: Text-prompt evolution v1 to v7. Each cluster of bars shows the macro purity of a single prompt version across the 10-reducer ablation. UMAP universally beats PCA; v7 IFC-aware prompts reach macro 0.740 with UMAP-8 and HDBSCAN.

Figure 5.6 plots the macro purity of every prompt version across the reducer ablation, and Table 5.4 lists the best result per version. The v1 baseline restricts the VLM to generic geometric vocabulary (primitive shape, proportion, surface and scale tags only, with no functional or IFC naming) and scores the lowest macro purity in the study, 0.534. Every subsequent version relaxes that restriction towards schema-aware language, and the macro purity climbs monotonically to 0.740 at v7. The single largest jump, v1 to v2 ($+0.135$), comes from allowing a free-text shape class and the OBB aspect ratio; the v7 instruction to use explicit IFC vocabulary adds the final $+0.018$.

5.4.1 The reducer ablation: UMAP universally beats PCA

A surprising finding from the 10-reducer ablation: UMAP at any $n_{\text{components}} \geq 4$ beats PCA-32 by $\approx$4 pp on text alone. The 768-d Gemini embeddings have substantial local neighbourhood structure that PCA's variance-maximising projection flattens out but UMAP's locality objective preserves. The single choice of swapping PCA for UMAP swung text macro from 0.696 to 0.740.

Table 5.4: Text-modality best result per prompt version.

Prompt version	Reducer	Algorithm	Macro purity
v1 (geometric/morphological tags)	UMAP-4	HDBSCAN	0.534
v2 (+geometry, aspect ratio)	UMAP-4	HDBSCAN	0.669
v3 (+thinness)	UMAP-4	HDBSCAN	0.674
v4 (compact natural language)	UMAP-4	HDBSCAN	0.720
v5 (natural names)	UMAP-4	HDBSCAN	0.715
v6 (+keywords)	UMAP-8	HDBSCAN	0.722
v7 (IFC-aware)	**UMAP-8**	**HDBSCAN**	**0.740**

5.4.2 Best-settings table

Table 5.5 fixes the prompt at the winning v7 and reports the best configuration of each clustering algorithm on the v7 text embeddings. HDBSCAN with leaf selection and a UMAP-8 reducer reaches macro 0.740 at $k = 128$. The fixed-k algorithms (K-Means, Bisecting K-Means, GMM) plateau near 0.64 at $K = 32$, roughly 10 pp behind, because they cannot reach the fine granularity at which the text signal separates the sub-types; PCA-32 is the best reducer for them, but UMAP-8 is uniformly better once HDBSCAN is free to choose its own k. The pattern is the same as on geometry: the algorithm that wins is the one that is allowed to over-cluster.

Table 5.5: Best settings for v7 text embeddings across algorithms.

Algorithm	Macro	k	Reducer	Settings
HDBSCAN	**0.740**	128	UMAP-8	leaf, mcs=5, ms=3, force
Bisecting	0.640	32	PCA-32	sklearn defaults
K-Means	0.637	32	PCA-32	sklearn defaults
GMM	0.624	32	PCA-32	full covariance

5.4.3 Where text wins and loses versus geometry

Text rescues identity-ambiguous types (Member, Plate, MEP) that have no distinctive geometric fingerprint, but loses on geometry-distinctive types (Column, Roof, Railing, Window) where the VLM description does not add information beyond what the OBB and orientation features already provide. This complementary pattern is exactly what we need for fusion to succeed.

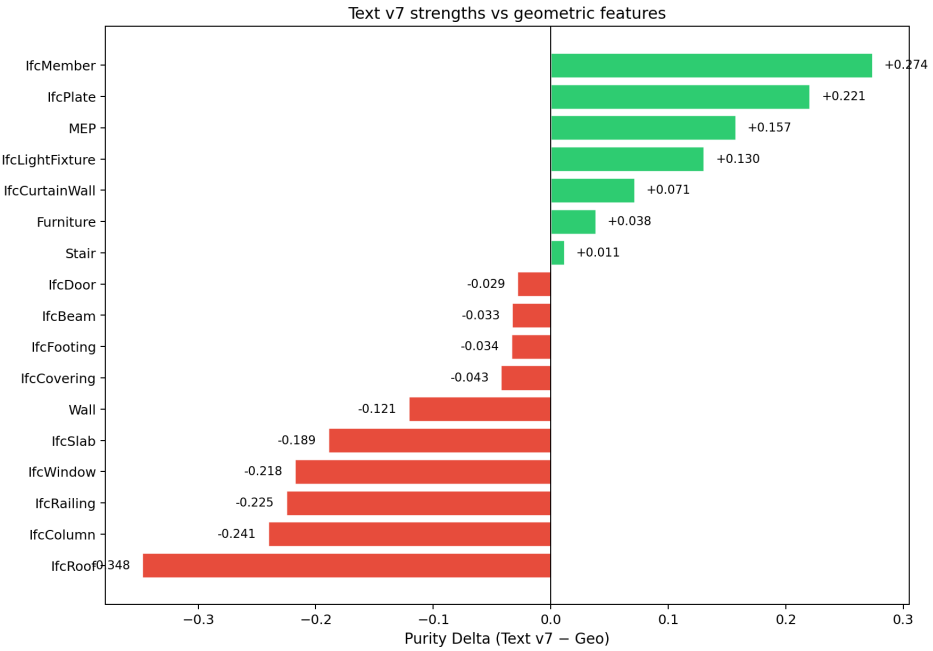


Figure 5.7: Per-type purity delta of text v7 versus the 9-feature geometric representation. Member (+0.274), Plate (+0.221), MEP (+0.157) and LightFixture (+0.130) all benefit from text; Column (−0.241), Roof (−0.348), Window (−0.218) and Railing (−0.225) suffer.

5.5 Visual Modality

We evaluate the three visual encoders across the reducer grid; Figure 5.8 compares them and Table 5.6 lists the best configuration per encoder. SigLIP2 on coloured renders with UMAP-16 leads at macro 0.779, with DINOv3 a fraction behind and all three encoders preferring UMAP over PCA over identity.

5.5.1 Per-type geometry-versus-vision delta

The complementary pattern repeats: vision is stronger on detailed, finely-featured types (Railing, MEP, LightFixture, Door) where the visual encoder picks up texture cues that geometry misses. Geometry remains the stronger signal for axis-extruded structural elements (Column, Beam, Roof), because their identity *is* essentially their PCA and OBB signature.

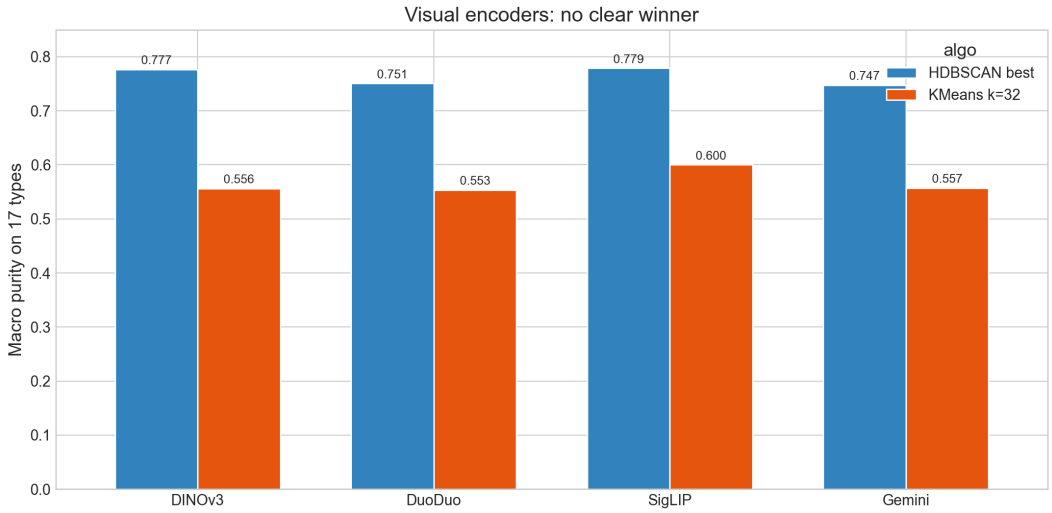


Figure 5.8: Visual encoders across reducers. SigLIP coloured with UMAP-16 reaches macro 0.779. DINOv3 follows at 0.777, DuoDuo CLIP at 0.751. All three encoders prefer UMAP over PCA over identity.

Table 5.6: Best results per visual encoder, on coloured renders.

Encoder	Variant	Macro	Reducer	Algorithm
SigLIP2	coloured	**0.779**	UMAP-16	HDBSCAN
DINOv3	coloured	0.777	UMAP-16	HDBSCAN
DuoDuo	coloured	0.751	UMAP-64	HDBSCAN
Gemini Embedding 2 (image)	coloured	0.747	UMAP-8	HDBSCAN

5.5.2 Coloured versus colourless

- SigLIP: coloured beats colourless by 0.7 pp.
- DINOv3: colourless beats coloured by 0.4 pp.
- DuoDuo: coloured beats colourless by 0.5 pp.

The effect is small and encoder-dependent, all within roughly 1 pp. We use the coloured renders for the visual modality: colour is part of what is geometrically and visually available for a BIM object, the strongest encoder (SigLIP) prefers it, and discarding it would throw away a freely available signal. The colourless variant is retained as

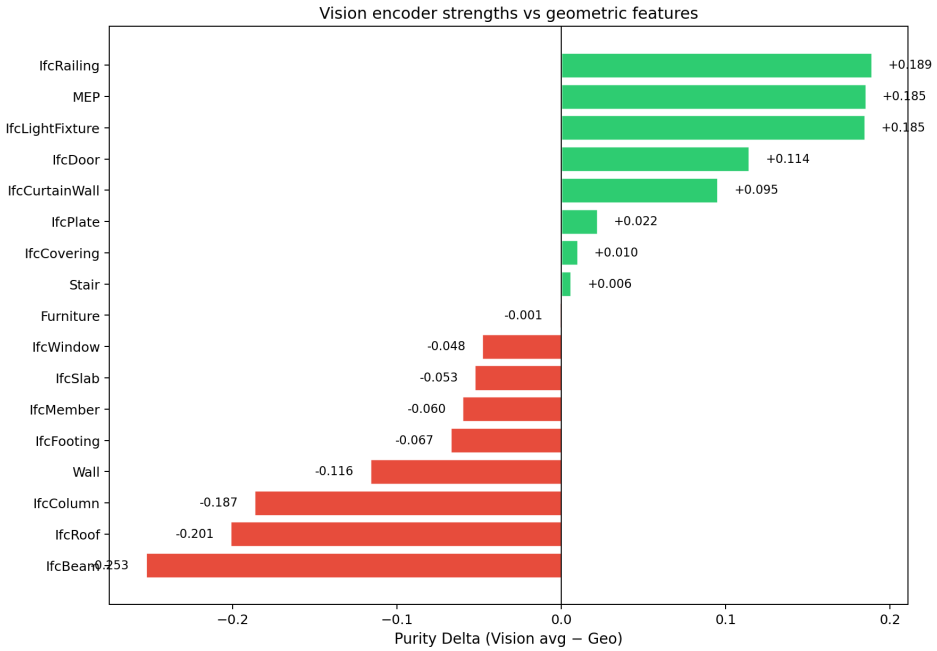


Figure 5.9: Per-type delta of vision (best coloured config) versus geometry. Vision rescues Railing (+0.189), MEP (+0.185), LightFixture (+0.185), Door (+0.114) and CurtainWall (+0.095); geometry remains stronger on Column (−0.187), Roof (−0.201) and Beam (−0.253).

a robustness check, and confirms that the pipeline still works when colour is absent, which matters because colour assignment is inconsistent across BIM authoring tools.

5.5.3 Per-type matrix across all five representations

Figure 5.10 collects all five single-modality representations into one per-type view. It confirms that no single modality dominates across types: geometry owns the structural, axis-extruded elements at the top, while the identity-noisy tail (Plate, Member) stays hard for every modality, which is precisely the gap multi-modal fusion is meant to close.

5.6 Multi-Modal Fusion

The full fusion sweep evaluates every recipe on the three modality subsets, for each visual encoder, over the 30-combination HDBSCAN grid. Table 5.7 reports the best

macro purity per recipe and subset.

Table 5.7: Fusion recipes per subset (best HDBSCAN configuration per cell, macro purity). F5_varbal is the dominant choice on every subset.

Subset	F1a	F1a_umap	F1b	F1b_varbal	F5	F5_varbal	F4 (Borda)
geo+text	0.785	0.799	0.722	0.795	0.702	**0.816**	0.739
geo+visual	0.826	0.826	0.748	0.806	0.697	**0.839**	0.794
geo+text+visual	0.812	0.819	0.769	0.825	0.736	**0.853**	0.793

F5_varbal wins every subset, and the pattern is consistent. Variance-balancing the modality blocks before a joint projection helps both reducers: it lifts joint PCA from F1b's 0.769 to F1b_varbal's 0.825, and joint UMAP from F5's 0.736 to F5_varbal's 0.853, an 11.7 pp swing in the latter case. Plain F5, without balancing, is in fact the *worst* recipe, because its neighbour graph is dominated by the high-dimensional text and visual blocks. The winner is therefore a global UMAP applied to a variance-balanced concatenation; the per-block recipes F1a and F1a_umap trail at 0.81–0.82 on the trimodal subset.

5.6.1 The headline winner

Table 5.8: Headline configuration: geo+text+visual via F5_varbal, the variance-balanced concatenation of the three blocks projected to four dimensions with UMAP, v7 text and SigLIP-coloured visual features, HDBSCAN at leaf, mcs=5, ms=3, with force-assign.

Quantity	Value
Macro purity (17 types)	**0.8530**
Overall purity	0.8729
Number of clusters k	126
Fused dimensionality	4 (UMAP on the variance-balanced concat)
Number of objects fused	1,700
Bootstrap ARI ($\pm$1 SD)	$\mathbf{0.874 \pm 0.019}$
Bootstrap macro purity	0.848 ± 0.008

Table 5.8 reports the headline configuration in full. Because macro purity rises with the cluster count k by construction (Section 4.9), the headline 0.853 is not read as a k-free quality score. Its credibility rests on three checks: at the matched $k = 126$ a plain

K-Means partition lands in the same range (Section 5.6.5), so the score is a property of the representation and the granularity rather than of HDBSCAN; the partition is stable under 90% resampling (bootstrap ARI 0.874); and the variance-balancing recipe transfers to a second corpus (Chapter 6). The auto-generated catalog (Section 5.12) is the qualitative counterpart, showing the clusters resolve into nameable component sub-types.

5.6.2 Why variance balancing makes F5_varbal win

The decisive comparison is F5 versus F5_varbal: the same global UMAP-on-concatenation projection, differing only in whether the three blocks are variance-balanced before concatenation. Plain F5 scores 0.736; F5_varbal scores 0.853, an 11.7 pp swing from a single re-scaling step. The reason is geometric. UMAP builds a k-nearest-neighbour graph in the concatenated space; with the raw standardised concat, the 768- and 1024-d text and visual blocks carry almost all of the Euclidean distance and the 9-d geometry block contributes negligible weight to neighbour selection, so the geometric signal is effectively discarded. Variance-balancing rescales each block to unit total variance, so geometry, text and visual contribute equally to the neighbour graph and UMAP can exploit all three. The same recovery holds across every encoder in the IFCNetCore audit (Chapter 6), where it lifts F1-macro by $+0.09$ to $+0.16$. It is the single most important architectural finding of the fusion study: *how* the blocks are balanced before a global projection matters more than the projection itself.

5.6.3 Are all three modalities pulling weight?

Comparing the F5_varbal winner across subsets:

- geo+text+visual (F5_varbal): 0.853
- geo+visual (F5_varbal): 0.839 (dropping text: -0.014)
- geo+text (F5_varbal): 0.816 (dropping visual: -0.037)

To test whether these gaps exceed sampling noise we run a paired object-level bootstrap on the three partitions, which are clustered on the same 1,700 objects and so resample together ($B = 2000$, with replacement). Adding the visual block to geo+text is significant: $\Delta_{\text{visual}} = +0.037$ (95 % CI $[+0.011, +0.050]$, $P(\Delta > 0) = 0.999$). Adding the text block to geo+visual is *not* significant at the 95 % level: $\Delta_{\text{text}} = +0.014$ (95 % CI $[-0.002, +0.036]$, $P(\Delta > 0) = 0.96$). The text gain is modest and statistically marginal partly because the Gemini v7 descriptions were generated *from* the rendered images, so text and visual share signal. The honest reading is that the working method is geometry

plus vision, with VLM text a third, partially-redundant signal whose contribution is positive on the point estimate but within bootstrap noise for clustering; as Section 5.13 shows, it carries a comparable, similarly small margin for retrieval.

5.6.4 Per-type purity at the fusion winner

The per-type breakdown at the fusion winner is shown in Figure 5.11. It achieves $\geq$0.9 per-type purity on 6 of 17 types (Stair 0.96, Footing 0.96, Railing 0.94, MEP 0.93, CurtainWall 0.93, Furniture 0.91) and at least 0.71 on every other type. Only Member (0.73), Plate (0.74), Window (0.76), Slab (0.80) and Door (0.71) sit in the harder band, and these are exactly the types whose human labelling is most subjective.

Read against the 9-feature geometric baseline, the per-type picture mirrors the modality deltas of Figures 5.7 and 5.9: fusion buys the most on the identity-ambiguous and visually-distinctive types that geometry alone cannot separate. Figure 5.12 shows the full F5_varbal $-$ geo delta. Member ($+0.32$), LightFixture ($+0.24$), MEP ($+0.22$), Railing ($+0.17$) and Plate ($+0.16$) gain the most, while the geometry-dominant types (Column, Beam, Roof) are unchanged (± 0.00) and only Window (-0.06), Wall (-0.04) and Slab (-0.01) regress marginally. No type is materially worse under fusion: the geometry signal is preserved while text and vision rescue the types it cannot reach.

5.6.5 K-matched comparison: HDBSCAN versus K-Means at the same k

The naive comparison in Section 5.3 pinned K-Means at $K = 32$ and HDBSCAN at $k \approx 100$. A *k-matched* comparison on the fusion winner (Table 5.9) clarifies how much of HDBSCAN's gap is partition quality versus granularity discovery:

Table 5.9: K-matched comparison on the F5_varbal fusion winner.

Method	k	**Macro Purity**	**Noise fraction**
K-Means (10 seeds)	126	$\mathbf{0.863 \pm 0.006}$	0%
HDBSCAN force=True	126	0.853	0%
HDBSCAN force=False	126	0.885 (assigned only)	20.1%
K-Means (10 seeds)	17	0.601 ± 0.031	0%

At matched $k = 126$, K-Means slightly outperforms HDBSCAN with force-assign by 1.0 pp; HDBSCAN's native partition (force=False) is purer (0.885) but dumps 20.1% of objects as noise. The honest reading is:

> *HDBSCAN's contribution at the fusion winner is* ***automatic granularity discovery*** *($k = 17 \rightarrow 126$ is a $+25$ pp macro jump on the same representation), not*

partition quality at a given k. Given that k, either HDBSCAN or K-Means delivers a deployment-quality partition on the F5_varbal representation.

We adopt HDBSCAN-force for the deployed catalog because it gives a complete no-noise partition in a single pass; the ~ 1 pp K-Means edge at matched k is acknowledged but does not change the deployment recipe. For the highest-confidence reusable-component subset, we additionally report HDBSCAN-no-force assigned-only clusters and treat the noise as "unclassified residue".

5.7 Supervised Ceiling

Table 5.10: Supervised Random-Forest ceiling per representation.

Subset	Macro recall	Accuracy
geo	0.878	0.893
text	0.721	0.774
visual	0.760	0.796
geo+text	0.884	0.902
geo+visual	**0.891**	**0.907**
text+visual	0.809	0.841
geo+text+visual	0.888	0.906

Observations:

- Random Forest is largely insensitive to text once geometry and visual are present ($+0.1$ pp from text+visual to geo+text+visual, -0.3 pp from geo+visual to geo+text+visual). The supervised setting cleanly identifies that text is redundant with visual.
- The per-modality ceilings in Table 5.10 are measured on a richer representation than the F5_varbal winner. Trained directly on the winner's own four-dimensional UMAP embedding, the Random Forest reaches macro recall 0.857 (5 seeds); on the 1801-dimensional variance-balanced concatenation that precedes the UMAP projection it reaches 0.853. Cluster purity and classifier recall are *different metrics* and should not be differenced, but the unsupervised macro purity of 0.853 on that same four-dimensional embedding sits right at its supervised ceiling: the clustering already recovers essentially all of the label-consistent structure the winning representation contains. The higher 0.888–0.891 per-modality ceilings locate the remaining headroom in the *representation*, not the clustering algorithm.

- PCA-8 per modality is enough: bumping to PCA-64 yields < 0.5 pp.

5.8 Stability and Sanity Checks

Table 5.11: Stability and sanity diagnostics.

Check	**Result**
Bootstrap ARI (45 pairs)	0.874 ± 0.019
Bootstrap macro purity	0.848 ± 0.008
K-Means at $k = 17$ (10 seeds)	0.601 ± 0.031
Δ_{visual} (paired boot.)	$+0.037$ CI $[+0.011, +0.050]$
Δ_{text} (paired boot.)	$+0.014$ CI $[-0.002, +0.036]$
Silhouette-picked $k^\star$	29 ($s = 0.253$)

The stability and sanity diagnostics are collected in Table 5.11. The bootstrap ARI of 0.874 is high for a density-based clustering at $k \approx 126$. This indicates the structure is genuinely present in the data, not an artefact of HDBSCAN's hyperparameters or of the subsample. All features are log-transformed where appropriate and standardised with a StandardScaler before clustering.

5.9 Per-Type Diagnostic Matrix

The matrix view in Figure 5.13 shows clearly which modality *rescues* which type:

- **Geometry-dominant types:** Stair (0.95), Wall (0.90), Footing (0.89), Column (0.87), Beam (0.86), Furniture (0.86), Covering (0.83), Slab (0.81), Roof (0.78), Railing (0.77).
- **Vision-dominant types:** Door, Railing, MEP and LightFixture, where SigLIP or DINOv3 alone outperforms geometry by 5 to 19 pp.
- **Hard tail:** Plate (geo 0.57, all-modality max 0.66) and Member (geo 0.42, all-modality max 0.42) remain stubbornly low even at the RF ceiling. These are the irreducible label-noise cases.

5.10 UMAP Visualisation of the Library

The UMAP-2D projection (Figure 5.14) is the most direct visualisation of the library structure recovered by the pipeline. Coloured by IFC type, it shows that type labels do form coherent regions in the fused embedding (the schema grain is preserved); coloured by the 126-cluster partition, finer-grained sub-groups emerge. Visual inspection of the resulting catalog (Section 5.12) confirms that the sub-clusters correspond to architecturally meaningful sub-types (operable double-leaf timber doors separating from single-leaf fire-rated doors, for example).

5.11 Cluster Sizes and Composition

The 126 clusters of the fusion winner are not uniform in size. Figure 5.15 shows the size distribution: the mode sits at 10–12 members per cluster, the smallest clusters are at the `min_cluster_size`=5 floor, and the largest holds a few dozen objects. The distribution has the long-tailed shape characteristic of density-based clustering: a body of mid-size clusters that each capture one coherent sub-type, plus a small number of larger clusters absorbing the most populous, internally varied classes (notably Furniture). No cluster degenerates into a catch-all: even the largest is dominated by a single IFC type. This size profile is what makes the partition usable as a library, since each cluster is small enough to inspect at a glance yet large enough to represent a recurring component rather than a one-off.

5.12 Auto-Generated Catalog

For each of the 126 clusters the catalog assembles one navigable entry, gathering: the cluster ID, member count, dominant IFC type and the type-composition breakdown; the three centroid-nearest exemplars, each rendered in all nine canonical views; the full member gallery; and the top-5 TF-IDF keywords mined from the Gemini v7 descriptions of the cluster members, for example `horizontal, extrusion, uniform, channel, member` for a railing sub-cluster. Figure 5.16 shows eight such entries. The catalog is generated entirely automatically: no domain expert was involved in authoring the keywords or grouping the members.

5.12.1 Library-meaningful sub-types

The library claim of the thesis title is concrete: each cluster should map to a recognisable component sub-type that the flat IFC label set cannot express. Table 5.12 samples twenty

100%-pure clusters from the catalog (filtering to clusters of size ≥ 10 for legibility), with the dominant IFC type and the top-5 TF-IDF keywords that the catalog builder mined automatically from the v7 descriptions of the cluster members.

Table 5.12: Twenty 100%-pure catalog clusters of size ≥ 10 (F5_varbal HDBSCAN, $k = 126$). Keywords are the top-5 TF-IDF tokens mined from the Gemini v7 descriptions of the cluster members, not authored by the thesis. *Seven* distinct Stair sub-types, *ten* distinct Furniture sub-types and *three* distinct Wall sub-types surface from a flat 17-class IFC schema.

#	size	dominant type	top-5 TF-IDF keywords
113	22	Furniture	desk, workstation, office, furniture, block section
42	20	Furniture	chair, seat, furniture, seating, block irregular
81	19	Wall	cutouts wall, scattered, vertical sheet, partition, panel
106	18	Footing	horizontal rod, extrusion beam, rectangular uniform, support medium
70	18	CurtainWall	curtain wall, glazed grid, window, medium vertical, facade panel
73	17	Furniture	table, furniture, circular, medium block, base
114	17	Column	vertical rod, extrusion column, post, uniform, member
3	17	MEP	toilet, fixture, plumbing, tank, bowl
123	17	Window	window, glazed, frame, vertical sheet, subdivided
27	16	Stair	treads, staircase, stepped, circulation, section
9	16	Railing	horizontal, extrusion, uniform, channel, member
61	15	Furniture	upholstered, seating, furniture, medium block, chair
41	15	Wall	scattered cutouts, partition panel, vertical sheet, perforated
47	14	Covering	panel slab, rectangular flat, ceiling, horizontal sheet, floor
64	14	Stair	treads, diagonal, stepped, staircase, sheet irregular
82	14	Wall	cutouts wall, scattered, vertical sheet, partition, panel
22	14	Roof	structural, beam, triangular, member, uniform extrusion
74	13	Stair	helical, spiral, stepped treads, staircase, stairs
96	13	Beam	channel, horizontal rod, uniform extrusion, member
36	13	LightFixture	lamp, vertical rod, lighting, floor, circular

The grouping is library-meaningful at the schema-grain level the IFC type cannot express: helical/spiral (cluster 74), straight-run (27) and diagonal-stringer (64) staircases all live under `IfcStair` but appear as distinct catalog entries; desk/workstation units (113), chairs (42), tables (73) and upholstered seating (61) all live under `IfcFurniture` but separate; walls with scattered cutouts (81, 41, 82) form their own family. These are exactly the reusable-component sub-libraries an architect or BIM curator would want to retrieve as a unit.

Table 5.13: Late-fusion retrieval recall@k on the 1,700-object benchmark.

Configuration	recall@1	recall@5	recall@10
geo only	0.848	0.941	0.960
text only	0.747	0.896	0.930
visual only	0.831	0.932	0.954
late geo+text	0.875	0.946	0.960
late geo+visual	0.874	0.950	0.963
late geo+text+visual	**0.884**	**0.952**	**0.966**

5.13 Late-Borda Retrieval

Figure 5.17 illustrates the late-Borda rank fusion and Table 5.13 reports recall@k across configurations, with the full geo+text+visual fusion reaching recall@10 $= 0.966$. Per-type recall@10 is ≥ 0.90 for all 17 types; three types (Stair, Furniture, CurtainWall) reach 1.000. The relevance ground truth used here is the *hand-annotated* type label (Section 4.2), not the noisy as-authored IFC type: the 1,700 objects were re-labelled directly during annotation, so a hit at rank k is a hit against the human-curated class, not against the raw exporter assignment. Retrieval is complementary to clustering: it uses the same three modalities and the same per-modality similarities, but combines them differently (rank fusion rather than feature fusion) for a different downstream objective (top-k ranking rather than partition).

5.14 Combined Discussion

5.14.1 Answering the research questions

RQ1 (geometry). The 4-feature OBB baseline is structurally inadequate (macro 0.434 at $k = 17$), but a targeted four-stage feature-engineering progression raises macro purity to 0.704 at $k = 32$, a $+0.197$ gain. Each stage was justified by an explicit cluster-confusion diagnosis at the previous stage. This shows that targeted, interpretable descriptors are a strong and lightweight geometry modality (under 0.5 s per object); we do not claim superiority over a learned point-cloud encoder, which we did not run on this corpus, only that compact handcrafted features are competitive enough to carry the geometry stream of the fusion.

RQ2 (text). VLM-derived descriptions, when carefully prompt-engineered (v1 generic geometric tags through v7 IFC-aware), reach macro 0.740 at $k = 128$ with UMAP-8 and HDBSCAN. Text alone is competitive with geometry alone, but the two modalities are

complementary: text rescues identity-ambiguous types (Member, Plate, MEP) where geometry struggles.

RQ3 (visual). General-purpose visual encoders are competitive: SigLIP coloured with UMAP-16 reaches 0.779, matching DINOv3 within 0.2 pp and beating DuoDuo by 2.8 pp. The reducer choice (UMAP $\gg$ PCA $\gg$ identity) matters more than the encoder choice ($<$1 pp gap among the top three). The honest takeaway: the encoder family is not the bottleneck.

RQ4 (fusion). F5_varbal, the variance-balanced concatenation of the three blocks projected with UMAP, wins every modality subset. The decisive lever is variance balancing: plain UMAP-on-concat (F5) is the *worst* recipe at 0.736 because the high-dimensional blocks dominate its neighbour graph, while balancing the blocks first lifts that same projection to 0.853. The per-block recipes (F1a, F1a_umap) trail at 0.81–0.82. The winning headline configuration is **F5_varbal geo+text+visual, HDBSCAN leaf mcs=5 ms=3 force,** $k = 126$**, macro 0.853**.

RQ5 (ceiling and stability). A Random Forest trained on the F5_varbal winning representation reaches macro recall 0.857, against the unsupervised macro purity 0.853 on that same representation; the two are different metrics, but their closeness shows the clustering already recovers essentially all of the label-consistent structure that representation contains, with the remaining headroom (a richer PCA-8 representation admits a 0.888–0.891 ceiling) being representational rather than algorithmic. Bootstrap ARI 0.874 ± 0.019 confirms the partition is stable under 90%-subsampling. The schema-grain constraint ($k = 17$) is far from the natural granularity of the data ($k \approx 126$).

5.14.2 Surprises and revisions to the first-pass narrative

Three findings revised our first-pass narrative substantially:

- **Variance balancing, not the projection, decides the fusion.** The best and worst trimodal recipes (F5_varbal 0.853 and F5 0.736) are the *same* UMAP-on-concat projection; the only difference is whether the blocks are variance-balanced first. Re-scaling each block to unit variance, so the 9-d geometry is not drowned by the 768- and 1024-d blocks, is worth 11.7 pp.

- **HDBSCAN's advantage is granularity, not partition quality.** At matched $k = 126$, K-Means slightly outperforms HDBSCAN-force. The story is no longer "HDBSCAN is better than K-Means" but "HDBSCAN finds the right k automatically", which is enormously valuable when the right k is unknown.

- **Text and visual are partially redundant.** Adding text on top of geo+visual adds +1.4 pp to macro purity, but a paired bootstrap puts that gain within noise (95 %

CI $[-0.2, +3.6]$ pp), whereas the visual block's $+3.7$ pp over geo+text is significant ($[+1.1, +5.0]$ pp). The Gemini v7 descriptions were generated from the rendered images, so the two modalities share signal. We carry text for the thesis-title commitment and for retrieval, but the honest fusion story is that the working method is geometry plus vision, with text a modest, statistically marginal third signal.

5.14.3 Limitations

- Cross-corpus generalisation is established on the public IFCNetCore benchmark in Chapter 6, but industrial-scale federated repositories remain future work.
- 17 merged classes may understate the schema grain in practice; an industrial dataset might have additional fine-grain distinctions.
- The Gemini API is closed-source; replacing it with an open VLM (LLaVA, Qwen-VL) is future work.
- HDBSCAN with force-assign is somewhat opinionated; the no-force variant produces purer clusters but with 20.1% noise.
- **Text-modality retrieval is weak on geometrically ambiguous objects.** When two objects of different IFC types share an OBB envelope and a generic shape (`IfcMember` versus `IfcPlate`, slender wall versus thin slab), the VLM description collapses to generic shape vocabulary and the text similarity carries little discriminative signal; these are the cases where the geometric and visual modalities have to do the work.

5.15 Summary

This chapter has reported the full set of experimental results: a 4-stage geometric feature progression lifting macro from 0.434 to 0.704; a 7-version text-prompt evolution to 0.740; a three-encoder visual bake-off to 0.779; a seven-recipe fusion ablation with F5_varbal winning at macro **0.853** (bootstrap 0.848 ± 0.008) with bootstrap ARI **0.874**; a supervised RF ceiling of macro recall 0.857 on that winning representation as an indicative upper bound; a k-matched comparison that re-frames HDBSCAN's advantage as granularity rather than partition quality; and two downstream applications, a 126-cluster catalog and late-Borda retrieval at **recall@10 = 0.966**. The next chapter validates these findings on a second corpus.

Figure 5.10: Per-type purity heatmap. Each row is one IFC type, each column one representation (geometry 9-d, DINOv3, DuoDuo, SigLIP, Gemini). Top rows (Stair, Wall, Footing, Column, Beam) are dominated by geometry. Bottom rows (Plate, Member) are hard for every modality: this is the identity-noisy tail.

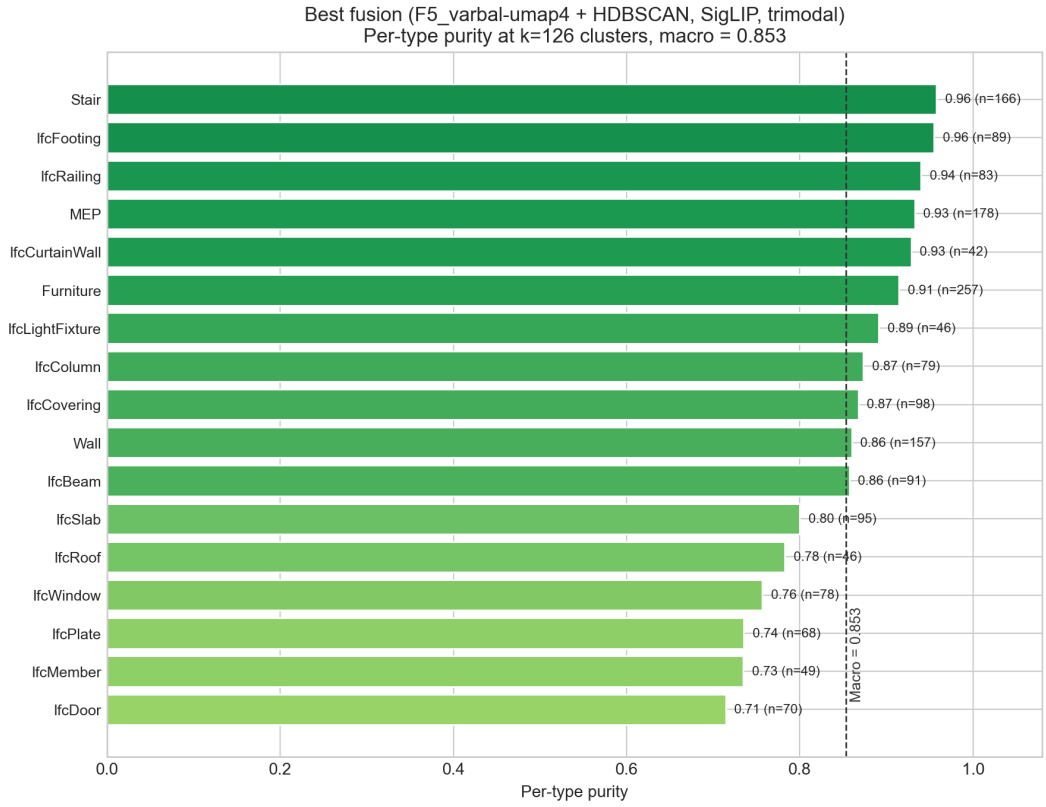


Figure 5.11: Per-type purity at the fusion winner ($k = 126$, F5_varbal geo+text+visual). 11 of 17 types reach ≥ 0.86; only Member, Plate, Door, Window and Roof sit below 0.80. Macro 0.853 on the 1,700-object benchmark.

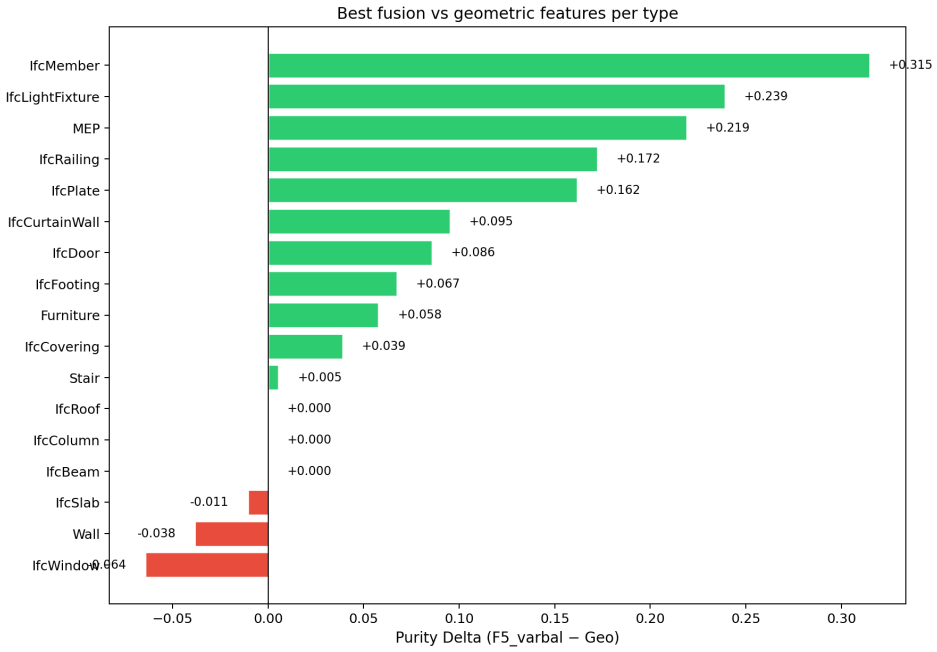


Figure 5.12: Per-type purity delta of the F5_varbal trimodal fusion winner versus the 9-feature geometric representation. Positive bars are types where fusion helps over geometry alone (Member +0.32, LightFixture +0.24, MEP +0.22, Railing +0.17, Plate +0.16); Column, Beam and Roof are unchanged and only Window, Wall and Slab regress marginally. Fusion adds where geometry is blind without sacrificing the types geometry already recovers.

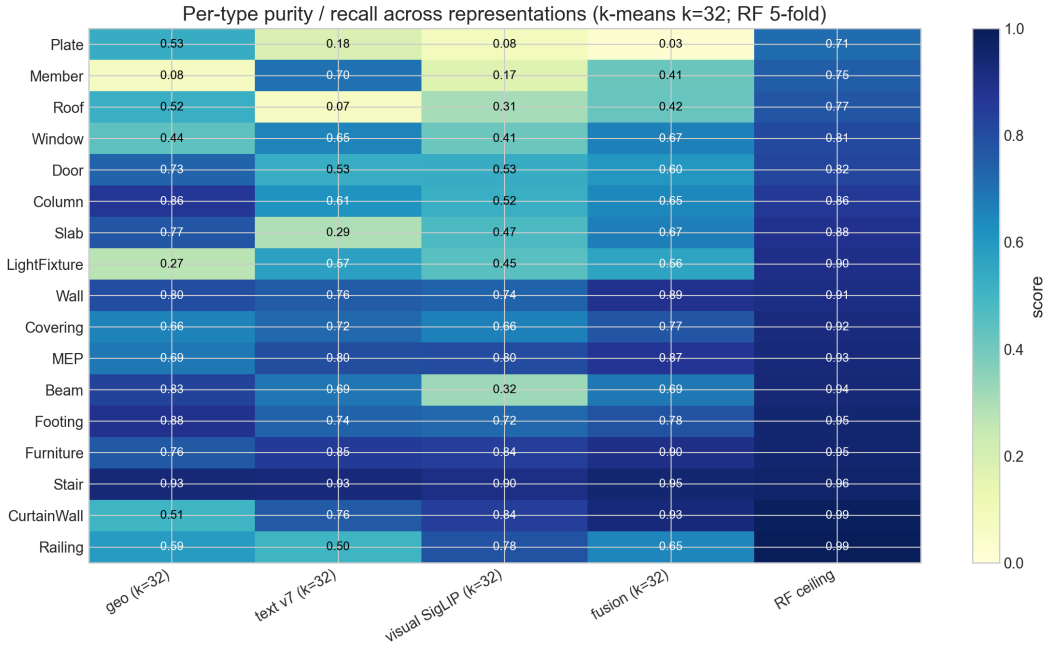


Figure 5.13: Per-type diagnostic. Rows: 17 IFC types. Columns: five representations (geometry, text v7, visual SigLIP, fusion, all at K-Means $k = 32$; and the Random-Forest 5-fold ceiling). Colour encodes per-type purity/recall. Each row's strongest cell identifies which modality is responsible for cleanly recovering that type.

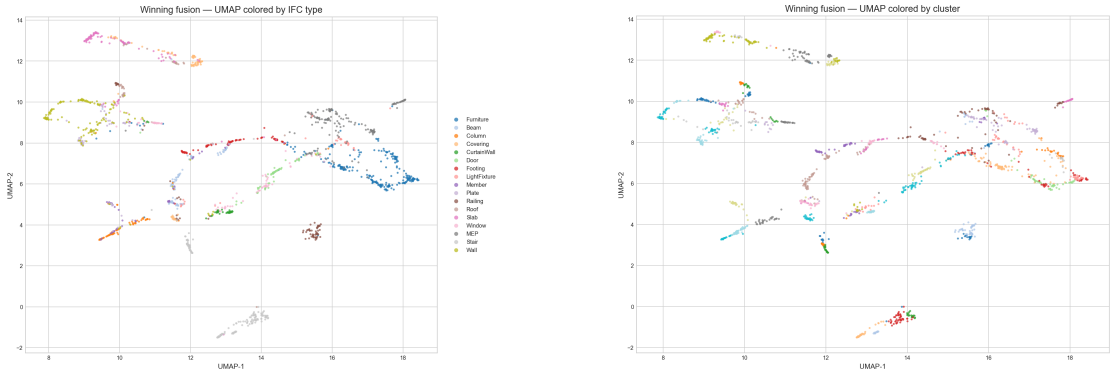


Figure 5.14: UMAP 2-D projection of the fused multi-modal embedding. Left: coloured by IFC type (17 colours), with type-coherent regions visible. Right: coloured by HDBSCAN cluster ($k = 126$), where finer sub-regions surface, many with intuitive interpretations (Furniture splits into chair / table / cabinet / shelf basins, Wall splits into interior / exterior / curtain).

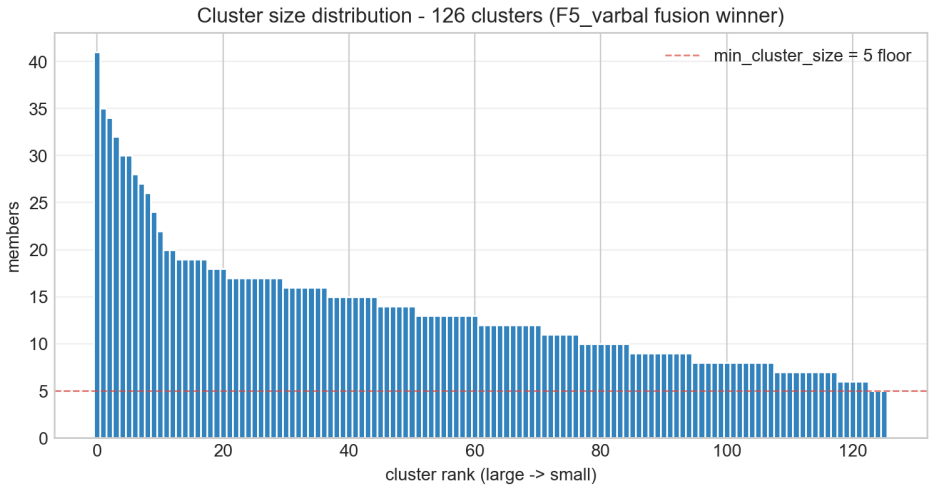


Figure 5.15: Cluster size histogram for the $k = 126$ fusion winner. The mode is at 10–12 members per cluster; the smallest clusters sit at the `min_cluster_size=5` floor; the largest holds a few dozen members.

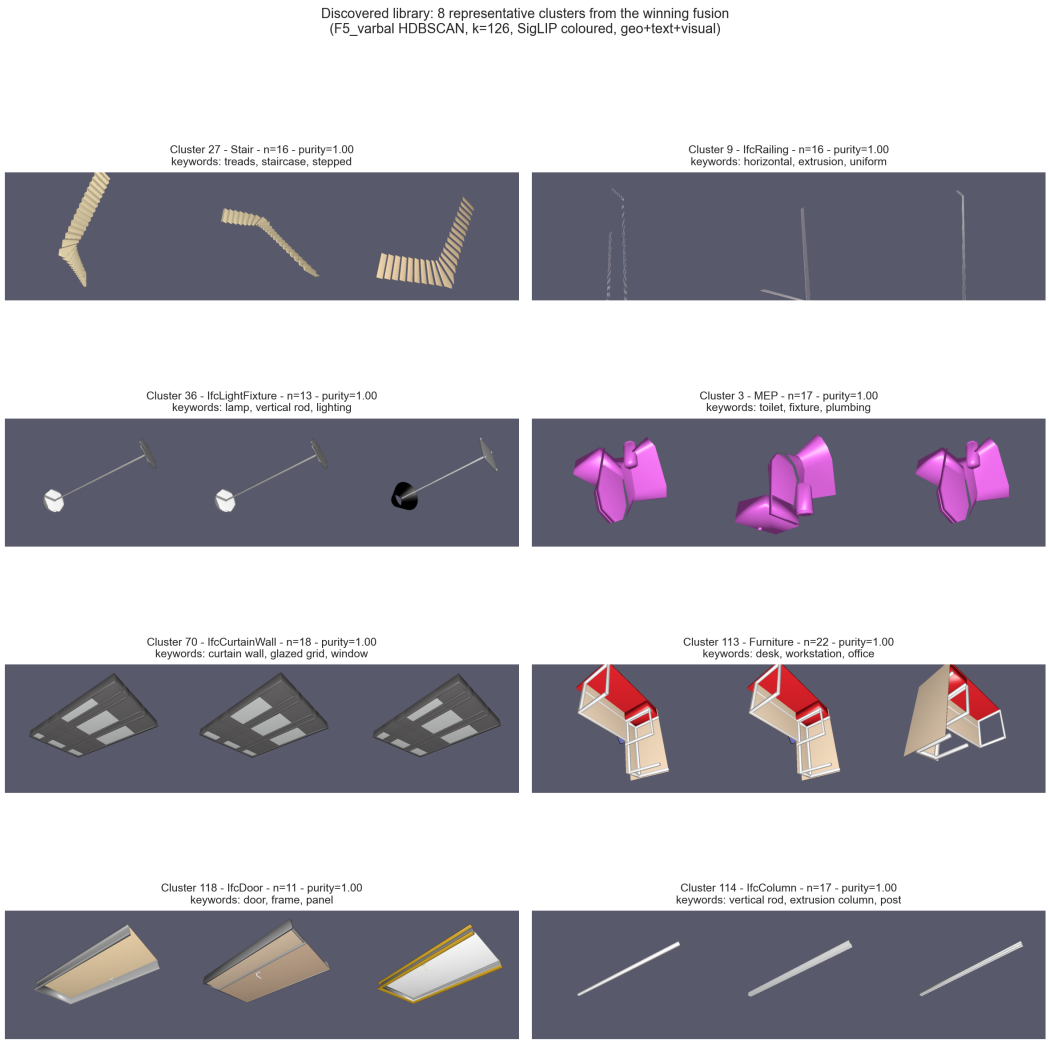


Figure 5.16: Eight representative clusters from the mined library. Each cluster is shown with three centroid-nearest exemplars and the TF-IDF keywords automatically extracted from the Gemini v7 descriptions of its members. The groupings (stair treads, railing infill, light fixtures, MEP terminals, curtain-wall panels, furniture, doors, columns) are produced without any manual labelling.

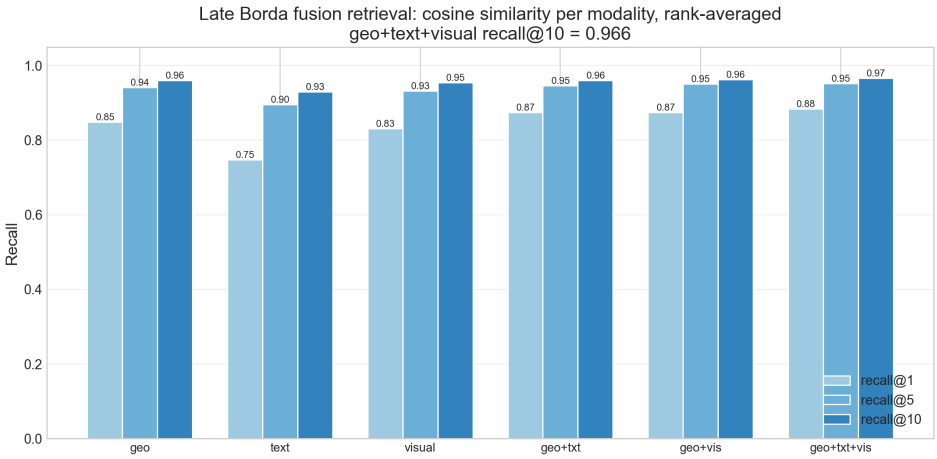


Figure 5.17: Late-Borda retrieval. Per-modality cosine similarity is rank-aggregated via Borda count; recall@10 on the 1,700-object benchmark reaches **0.966** for the full geo+text+visual fusion.

6 Cross-Corpus Validation on IFCNetCore

Chapter 5 established the headline results of the thesis on the Bentley/TUM corpus: 1,700 manually annotated objects across 17 merged IFC classes derived from 202 student submissions. A natural concern that immediately arises is *external validity*: do the engineered features, the fusion recipes and the algorithm choices generalise to a different IFC modeller distribution and a different (finer) class taxonomy, or are they fitted to the idiosyncrasies of the TUM student corpus?

This chapter answers that question by re-running the full pipeline on the public **IFCNetCore** benchmark [16]: 7,930 objects across 20 native IFC classes with an official train/test split. Every number quoted in this chapter is backed by a persisted results file, and every figure is regenerable from the experiment scripts.

6.1 Why IFCNetCore?

The Bentley/TUM corpus is internal: 1,700 objects, 17 merged labels, drawn from 202 student submissions, with some classes present only because of the type-merge step. IFCNetCore by contrast is:

- **Public**, released by Emunds et al. in 2021.
- **Roughly** 5× **larger** (7,930 vs. 1,700 objects).
- **Finer-grained**, with 20 native IFC classes and no merging.
- **Pre-split**, with an official train/test partition (≈70/30), enabling clean supervised evaluation under the benchmark's own ground rules.
- **Balanced in supervision**, with a long tail (`IfcStair` 52 objects, `IfcOutlet` 59, `IfcLamp` 92) but no class that disappears.

The class distribution and the official train/test split are shown in Figure 6.1. Running the same recipes on this distribution asks two concrete methodological questions:

1. **Do the 9 handcrafted geometric features generalise?** Either they re-establish a comparable supervised ceiling on a different IFC distribution, or they overfit the Bentley statistics and collapse.

2. **Do the fusion recipes (F1a, F5_varbal and the rest) hold their relative ranking?** If F1a wins on both corpora and F5_varbal is the unsupervised winner on both, the recipe ranking has external validity; if rankings reshuffle, the Bentley-specific results were lucky.

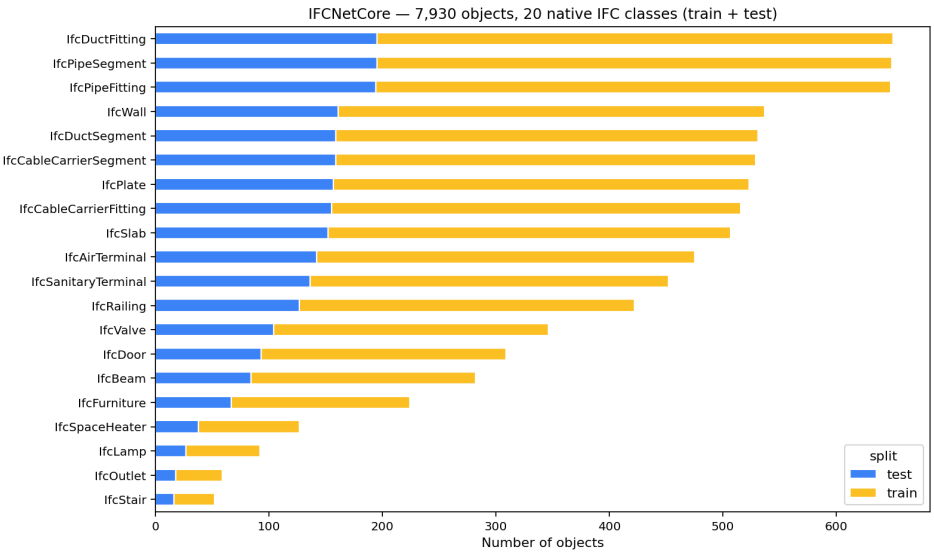


Figure 6.1: IFCNetCore class distribution. 20 native IFC classes, official train/test split ≈ 70/30. Long-tail classes include `IfcStair` (52), `IfcOutlet` (59) and `IfcLamp` (92).

6.2 Pipeline Parity via `DatasetSpec`

To re-use every line of code from the Bentley experiments without copying, the data-loading layer was refactored around a `DatasetSpec` dataclass that captures the four things that differ across corpora:

- `mesh_dir`: where the OBJ files live.
- `key_col`: the column used as the object identifier (`GlobalId` on Bentley, `obj_id` on IFCNet).
- `label_col`: the column carrying the ground-truth class.
- `train_test_col`: the column carrying the split indicator (`NaN` on Bentley, `{train,test}` on IFCNet).

A single environment switch (`IFCNET_DATASET=ifcnet`) retargets every downstream consumer (the feature-extraction and experiment scripts). The notable additions are:

- A dedicated baseline-extraction step: IFCNet has no IFC-schema parquet (the mesh OBJ files were pre-extracted by the benchmark authors), so we derive the Length / CSA / Volume / NumVertices baseline directly from `trimesh.bounding_box_oriented`.
- A generated metadata table built from the unzipped tree of 7,930 OBJ files. Object IDs are globally unique 32-character hex hashes; no class prefix is needed, since the label is carried by the parent directory in the IFCNetCore distribution.

The IFCNet render side comes pre-packaged: 12 grey-scale renders per object (silhouette-style), so on this corpus we target only the *colourless* variant codepath.

6.3 Geometry-Only Ceiling on IFCNet

The same 9-dimensional handcrafted feature vector, namely four log-OBB features (`Length, CSA, Volume, NumVertices`) plus five engineered features (`pc1_z, pc3_z, cs_aspect_ratio, normal_entropy, horiz_frac`), is computed on the 7,930 IFCNet objects without any retraining or re-tuning.

6.3.1 Unsupervised

Table 6.1: Geometry-only unsupervised clustering on IFCNetCore.

Algorithm	**Best config**	**Macro purity**	k
K-Means @ k=20	matches class count	0.308	20
K-Means @ k=48	overcluster 2.4×	0.446	48
HDBSCAN	`mcs`=5, leaf, force-assign	**0.733**	511
HDBSCAN	`mcs`=15, eom, force	0.574	113

The unsupervised geometry-only results are reported in Table 6.1. The headline 0.733 macro purity is reached at $k = 511$, a substantially fragmented partition (16 objects per cluster on average against 20 ground-truth classes). The honest operating point is HDBSCAN at `mcs`=15, $k \approx 113$, macro 0.574. Both numbers re-confirm the Bentley pattern: the schema grain (20 classes) is much coarser than the data-natural granularity that HDBSCAN discovers.

6.3.2 Supervised

Table 6.2: Geometry-only supervised ceiling on IFCNetCore (Random Forest 400 trees, train→test honouring the official split, 5 seeds).

Metric	Mean ± std
Accuracy	0.871 ± 0.001
F1-macro	$\mathbf{0.857 \pm 0.003}$
F1-weighted	0.871 ± 0.001
Macro recall	0.844 ± 0.003
Macro precision	0.876 ± 0.003

This is the key generalisation check, reported in Table 6.2. Applied to IFCNet's 20 native classes with no retuning, the 9 handcrafted geometry features reach a supervised **F1-macro of** 0.857 (macro recall 0.844). The Bentley supervised geometry ceiling, on 17 merged classes (Chapter 5, Table 5.10), is macro recall 0.878. The features therefore re-establish a comparable supervised ceiling on a corpus they were never tuned on, so the unsupervised gap on IFCNet ($\sim$0.4 macro purity at $k = 20$) is a *taxonomy-granularity* effect, not a *feature-quality* one.

6.3.3 Sanity check: MEP-family collapse

A concrete proof of the granularity claim: IFCNet contains eight MEP-family classes (`IfcPipeSegment`, `IfcPipeFitting`, `IfcDuctSegment`, `IfcDuctFitting`, `IfcCableCarrierFitting`, `IfcCableCarrierSegment`, `IfcCableSegment`, `IfcCableFitting`). Collapsing all eight into a single "MEP" superclass flips K-Means@$k = 20$ *overall* purity from 0.392 to 0.747. The granular MEP subcategories are geometrically very similar; the schema distinguishes them by function rather than shape. Figure 6.2 makes this concrete per class: seven classes receive 0% unsupervised purity yet are recovered by the supervised Random Forest.

6.4 Vision Encoder Extraction

For the visual modality we re-run all three encoders on the IFCNet colourless renders, using one Colab notebook per encoder, each of which also has a cluster-runnable script equivalent:

Table 6.3 lists the backbone, embedding dimension and extraction wall-time for each encoder. The aggregation matches Chapter 4: DINOv3 and SigLIP2 emit one

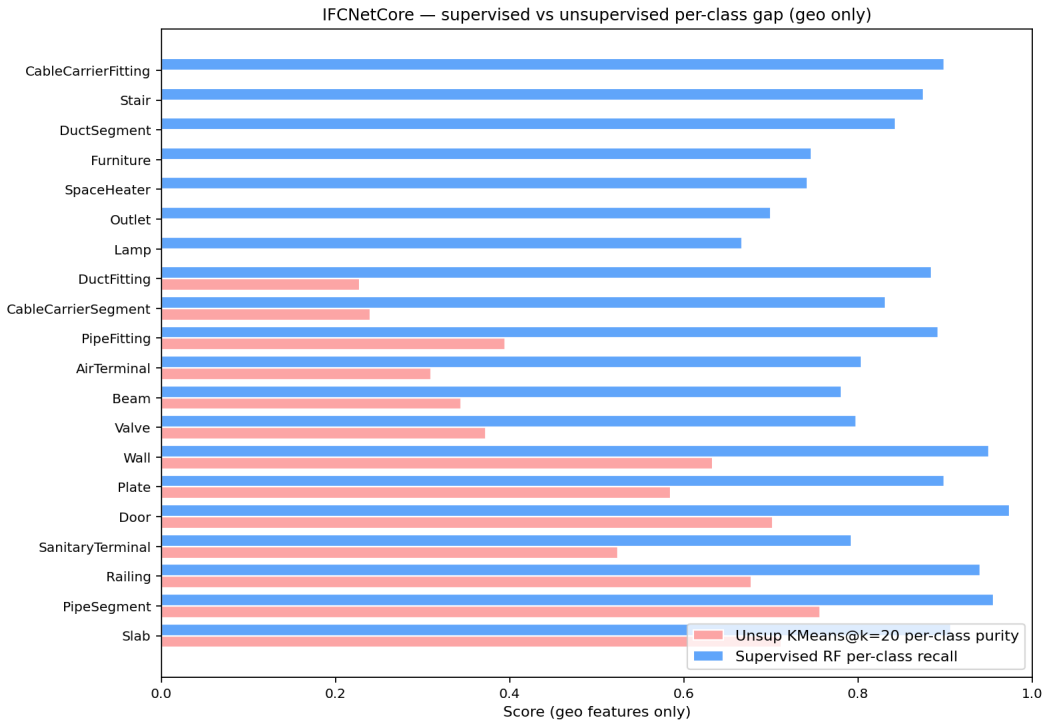


Figure 6.2: Per-class gap between supervised RF recall (blue) and unsupervised K-Means@$k=20$ purity (red). Seven classes receive exactly 0% unsupervised purity (absorbed into MEP-family clusters) but RF recovers them via non-linear boundaries. These are the classes where vision and text fusion should pay off.

embedding per view and are mean-pooled across the 12 IFCNet views, while DuoDuo CLIP takes all views at once and pools them internally. Each extraction script is resume-safe: an object whose embedding is already on disk is skipped. The cached embeddings follow the same per-encoder, per-variant naming convention as the Bentley pipeline, so no loader changes were needed.

6.5 Vision-Only Baselines

The vision-only supervised and unsupervised results are compared in Figure 6.3, with the supervised numbers tabulated in Table 6.4. SigLIP wins single-modality, consistent with its language-aligned contrastive training on a larger image–text corpus. DINOv3 (self-supervised, vision-only) is slightly behind. Vision-only *unsupervised* macro purity

Table 6.3: Visual encoder extraction on IFCNetCore. DINOv3 and SigLIP2 emit one embedding per view (mean-pooled over the 12 IFCNet views); DuoDuo CLIP pools the views internally.

Encoder	Backbone	Dim	Wall-time (T4)
DuoDuo CLIP	public 3D-tuned release	512	∼1.5 h
DINOv3 ViT-L/16	ViT-L/16, LVD-1689M	1024	∼3.5 h
SigLIP2	siglip2-large-patch16-512	1024	∼1.2 h

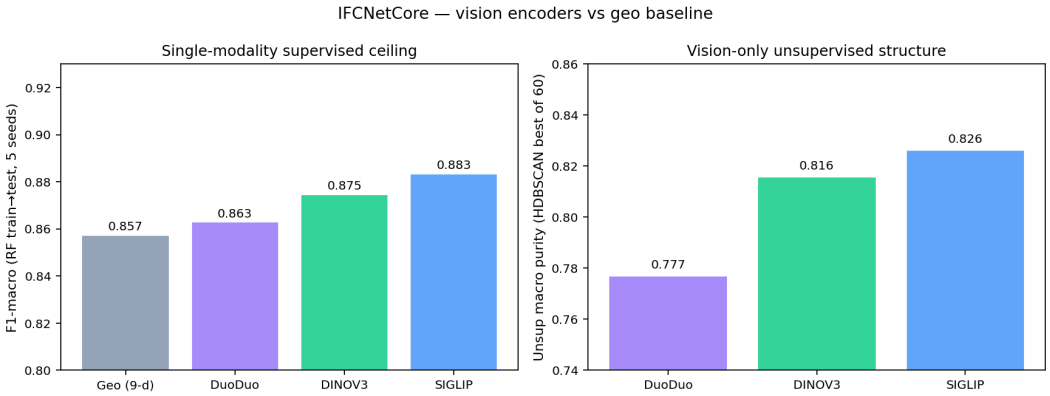


Figure 6.3: Vision-only encoder comparison on IFCNetCore. Left: supervised F1-macro. Right: unsupervised macro purity. The geo baseline (grey) is shown for reference on F1.

(HDBSCAN over the 30-combination grid) is DuoDuo 0.777, DINOv3 0.816, SigLIP 0.826: the same SigLIP $>$ DINOv3 $>$ DuoDuo ranking, with absolute numbers comparable to the Bentley visual-only results.

6.5.1 Geometry versus vision complementarity

Per-class diagnosis confirms the complementary pattern observed on Bentley:

- **Vision wins on texture-rich classes:** `IfcFurniture` (+0.20 F1 over geometry), `IfcSanitaryTerminal` (+0.14), `IfcCableCarrierFitting` (+0.05), and `IfcStair` reaches a perfect 1.000 under vision.
- **Geometry wins on shape-prior-heavy classes:** `IfcWall`, `IfcPlate`, `IfcSpaceHeater`.

This is exactly the structure that makes fusion deliver lift rather than averaging: the two modalities make different errors.

Table 6.4: Vision-only RF (400 trees, train→test, 5 seeds) on IFCNetCore.

Encoder	dim	Acc	F1-macro	Recall-macro	Δ vs. geo
DuoDuo	512	0.878	0.863	0.846	+0.006
DINOv3	1024	0.889	0.875	0.859	+0.018
SigLIP2	1024	**0.890**	**0.883**	**0.867**	+0.026

6.6 Per-Encoder Fusion Sweep (7 Strategies)

The same fusion recipes evaluated on Bentley are re-run on IFCNet for each of the three encoders; the 7 strategies are listed in Table 6.5, and the full supervised and unsupervised sweep is shown as a heatmap in Figure 6.4.

Table 6.5: Fusion strategies evaluated per encoder on IFCNetCore. 7 strategies × 3 encoders × {supervised RF, HDBSCAN 30-combination grid}.

Strategy	Recipe
F1a	per-modality PCA-16, concat with raw geo (d=25)
F1a_umap	per-modality UMAP-16, concat with raw geo (d=25)
F1b	std-concat, then PCA-16
F1b_varbal	varbal-concat, then PCA-16
F5	std-concat, then UMAP-16
F5_varbal	varbal-concat, then UMAP-16
CCA	CCA between geo and (text ⊕ visual) at d=8

6.6.1 Per-encoder winners

Table 6.6: Per-encoder winning fusion recipe on IFCNetCore.

Encoder	Sup winner	Sup F1-macro	Unsup winner	Unsup macro purity
DuoDuo	F1a	**0.912**	F5_varbal	0.818
DINOv3	F1a_umap	0.918	F5_varbal	**0.860**
SigLIP2	F1a / F1a_umap	**0.922**	F5_varbal	0.858

The per-encoder winning recipes are summarised in Table 6.6, and the cross-encoder findings reduce to three claims:

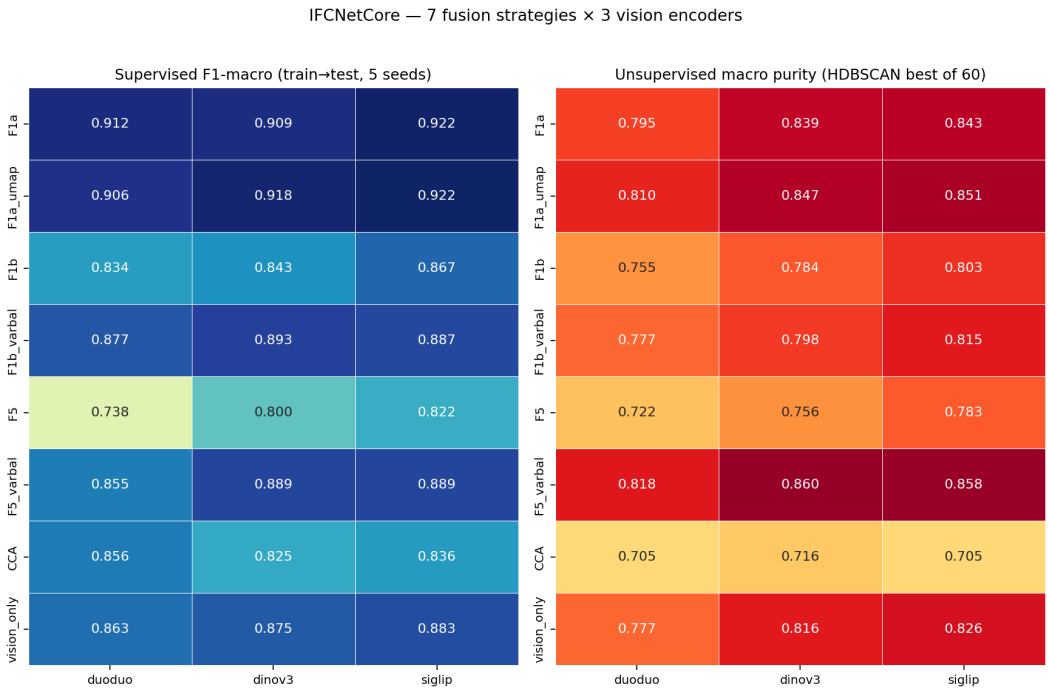


Figure 6.4: Full fusion heatmap. 7 fusion strategies plus a vision-only row (rows) × 3 encoders (columns). Left: supervised F1-macro. Right: unsupervised macro purity (HDBSCAN best of 60).

1. **F1a / F1a_umap is the most robust supervised recipe.** It is top-2 on all three encoders and beats every alternative by 0.02 to 0.10 F1.

2. **F5_varbal is the most robust unsupervised recipe.** It is top-1 unsupervised across all three encoders. DINOv3 with F5_varbal reaches 0.860 macro purity, the single highest unsupervised score in the entire study (across both datasets).

3. **The variance-balancing recovery is universal.** The same effect is observed on Bentley and across all three IFCNet encoders, as quantified in Figure 6.5:

6.7 Text Modality on IFCNet: Gemma 4 and EmbeddingGemma

The Bentley pipeline's text modality uses two closed-source Google models: Gemini 3.1 Flash Lite to generate the descriptions and Gemini Embedding 2 to embed them. Running Gemini 3.1 Flash Lite across the 7,930 IFCNet objects, and then re-running

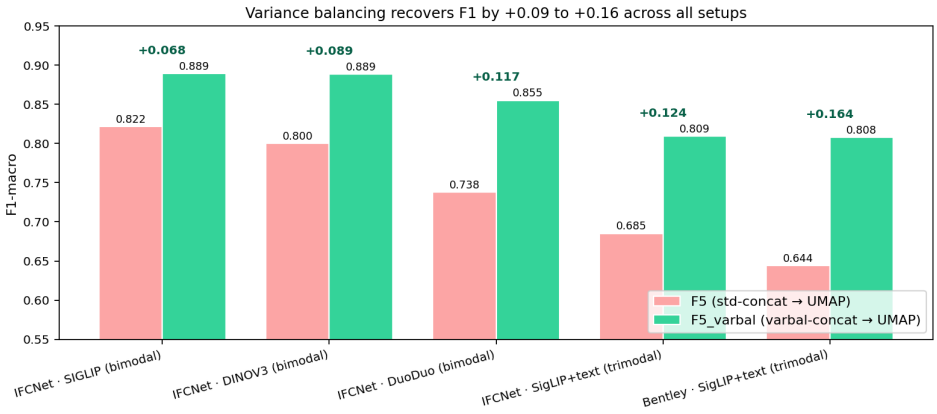


Figure 6.5: F5 (UMAP on std-concat) versus F5_varbal (UMAP on variance-balanced concat) F1-macro across 5 setups. Re-scaling the geo block by $1/\sqrt{\Sigma \text{var}}$ before UMAP-on-concat, so that it stops being dominated by the 768/1024-d embedding block, recovers F1 by $+0.09$ to $+0.16$ in every configuration: a free lunch from one line of math.

Bentley for a like-for-like comparison, would have multiplied the API cost beyond the budget for this thesis. We therefore switched both corpora to the open Gemma family for the cross-corpus comparison: **Gemma 4** generates a structured description of each object from its renders, and **EmbeddingGemma** embeds that description into a vector. EmbeddingGemma is a compact open embedding model whose 768-d output matches the dimensionality of Gemini Embedding 2 exactly, so the text block enters the fusion recipes unchanged. Because both corpora are re-described and re-embedded with the same open pair, the Bentley-vs-IFCNet text comparison is on a fully matched footing rather than a closed-vs-open one.

Each object is described once and embedded into a single per-object vector. Using open models here is deliberate: the IFCNet validation does not depend on a proprietary API, and it is a first step towards the open-VLM pipeline discussed in Section 7.2.

6.8 Ceiling Progression: Geo, Vision, Bimodal, Trimodal

Figure 6.6 and Table 6.7 trace the supervised F1-macro ceiling across tiers: it climbs $+0.065$ from geometry-only to the best geo+SigLIP bimodal fusion, but adding the open-model text modality does not push past that bimodal ceiling.

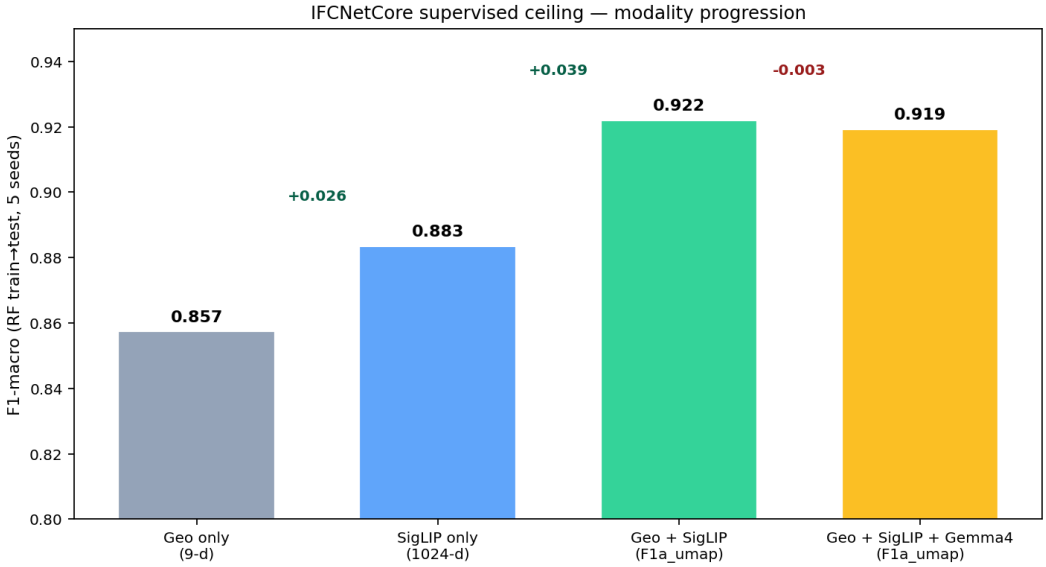


Figure 6.6: Ceiling progression on IFCNetCore. Supervised F1-macro climbs +0.065 from geometry-only (0.857) to the best bimodal fusion (0.922). Adding the Gemma 4 / EmbeddingGemma text modality (trimodal) does *not* push past the bimodal ceiling.

6.8.1 Why does text not help on IFCNet?

The most likely explanation is *prompt overfitting to the Bentley distribution*. The v7 system prompt was iteratively refined against the per-class confusion patterns of the Bentley 17-class taxonomy (Chapter 4, Section 4.5); its vocabulary hints, shape categories and disambiguation cues are tuned to surface the distinctions that matter *there*. Applied unchanged to IFCNet's 20 native IFC classes, whose label structure (granular MEP family, generic `IfcPipeSegment` / `IfcDuctFitting` names, no `Furniture` or `MEP` supercategory) does not match the Bentley distinctions the prompt was steered for, the descriptions become less informative than the contrastively aligned SigLIP image embedding.

6.9 Comparison with Published IFCNet Baselines

The geometry-only and bimodal numbers established above can be placed directly against the IFCNet paper's own deep-learning baselines on the same train/test split. Emunds et al. [16] evaluate MVCNN [17], DGCNN [18] and MeshNet [19] on IFCNet-

Table 6.7: Supervised F1-macro ceiling progression on IFCNetCore.

Tier	Recipe	F1-macro
Geometry only (9-d)	RF on standardised X	0.857
SigLIP only (1024-d)	RF	0.883
Geo + SigLIP bimodal	F1a / F1a_umap (d=25)	**0.922**
Geo + SigLIP + Gemma 4 text trimodal	F1a_umap (d=41)	0.919

Core; their follow-up, SpaRSE-BIM [20], introduces a sparse convolutional network on the same benchmark. Table 6.8 places the published numbers beside the present work.

Table 6.8: Published IFCNetCore baselines versus the handcrafted-feature and fusion results of this chapter. Published values are reproduced from the cited papers as macro-averaged F1; the train/test split is the official IFCNetCore partition shared across all rows.

Method	Modality	F1-macro
MVCNN [16, 17]	Multi-view CNN	0.869
SpaRSE-BIM [20]	Sparse voxel	≈ 0.818
9-feature RF (this work)	Handcrafted geometry	0.857
SigLIP2 + RF (this work)	Foundation-model vision	0.883
Geo + SigLIP bimodal F1a / F1a_umap (this work)	Geo + vision fusion	**0.922**

Two observations follow. First, the nine handcrafted geometric features alone, plugged into a Random Forest with no deep learning or backbone training, are competitive with the published deep baselines on F1-macro and exceed the sparse-voxel SpaRSE-BIM number reported by the same group. Second, the geo+SigLIP bimodal fusion lifts F1-macro by approximately 5 pp above the strongest published single-modality baseline (MVCNN) on the identical train/test split, locating the remaining performance gap in the choice of representation rather than the choice of classifier.

6.10 Cross-Dataset Trimodal Ranking

Figure 6.7 plots the trimodal strategy ranking on both datasets, and Table 6.9 reports the matched F1-macro values. Spearman rank correlation across the 7 strategies is $\rho \approx 0.93$. **The fusion-recipe ordering generalises.**

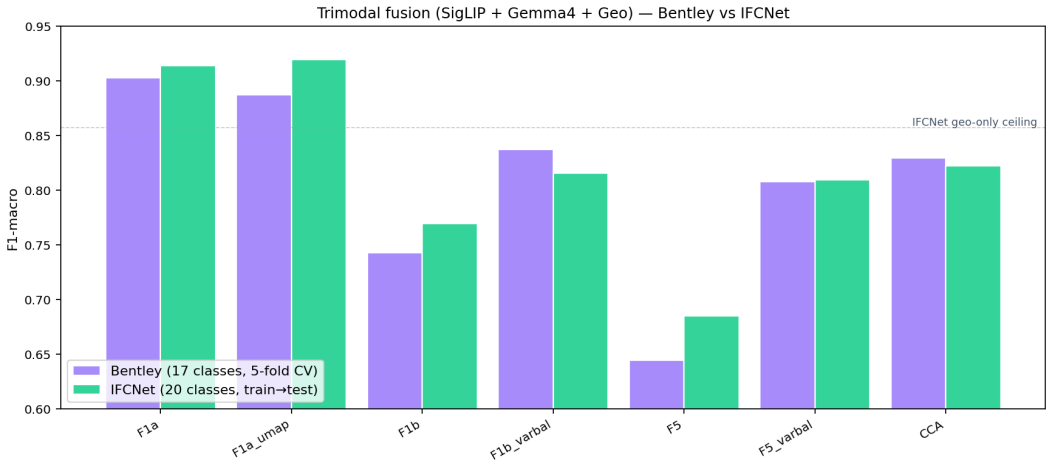


Figure 6.7: Trimodal fusion (SigLIP + text + geo) on both datasets, identical recipes. Strategy ranks are preserved across datasets; absolute numbers on IFCNet are higher partly because classifying 20 fine classes with balanced supports is intrinsically easier than 17 merged classes with long-tail supports.

6.11 Key Findings of the Cross-Corpus Audit

1. **Geometry features generalise.** Applied to IFCNet with no retuning, the 9 features reach supervised F1-macro 0.857 (macro recall 0.844) on 20 native IFC classes, a ceiling comparable to the Bentley supervised geometry result.

2. **Vision and geometry are complementary, not redundant.** Vision wins texture-rich classes (Furniture, Sanitary, Stair); geometry wins shape-prior-heavy classes (Wall, Plate, SpaceHeater). Fusion lift is $+0.04$ to $+0.05$ F1-macro over either modality alone.

3. **SigLIP > DINOv3 > DuoDuo on IFCNet supervised**, but **DINOv3 wins unsupervised** with F5_varbal at 0.860 macro purity (the highest unsupervised score in the entire study).

4. **F1a / F1a_umap is the universal supervised winner.** It holds across Bentley and IFCNet, and across DuoDuo, DINOv3 and SigLIP.

5. **F5_varbal is the universal unsupervised winner.** Same pattern.

6. **Variance balancing is a free lunch.** Adding $/\sqrt{\Sigma \text{var}}$ to each block before UMAP/PCA-on-concat recovers F1 by $+0.09$ to $+0.16$ at zero cost.

Table 6.9: Trimodal fusion (SigLIP + text + geo) cross-dataset F1-macro.

Strategy	Bentley (5-fold CV)	IFCNet (train→test)
F1a	0.903	0.914
F1a_umap	0.887	**0.919**
F1b_varbal	0.837	0.815
CCA	0.830	0.822
F5_varbal	0.808	0.809
F1b	0.743	0.769
F5	0.644	0.685

7. **Gemma 4 text adds nothing on IFCNet trimodal** (0.919 vs. 0.922 bimodal), against a slight bump on Bentley (+0.005 macro recall).

6.12 Reproducibility

The IFCNet experiments reproduce end-to-end as a fixed sequence of phases: building the object metadata; extracting the baseline OBB and the nine engineered geometric features; running the geometry-only unsupervised sweep and supervised ceiling; extracting the three vision encoders on Colab and caching their embeddings; running the per-encoder fusion sweep and vision-only Random Forest; running the trimodal fusion on both corpora; and finally regenerating the figures. The whole sequence is driven by a single dataset switch, so the identical code runs on both the Bentley and IFCNet corpora.

Each step is resume-safe: an incremental summary is written after every strategy and object, and re-running skips what is already on disk.

6.13 Summary

This chapter has re-run every modality, every fusion recipe and the supervised ceiling on the public IFCNetCore benchmark (7,930 objects, 20 native IFC classes). The headline conclusions are:

- The 9 handcrafted geometric features reach a supervised F1-macro of 0.857 on IFCNet's 20 native classes without retuning, a ceiling comparable to their Bentley performance and confirmation that the feature engineering was not over-fitted to the Bentley distribution.

- The fusion-recipe ranking generalises (Spearman $\rho \approx 0.93$ across 7 trimodal strategies). F1a / F1a_umap is the universal supervised winner; F5_varbal is the universal unsupervised winner.
- Variance balancing recovers F1 by $+0.09$ to $+0.16$ across every UMAP-on-concat configuration on every encoder, a free lunch we recommend as standard practice.
- Gemma 4 text adds nothing on IFCNet trimodal, in contrast to its slight positive contribution on Bentley.

With external validity established, the next chapter draws the overall conclusions and outlines the future work.

7 Conclusion and Future Work

7.1 Conclusion

This thesis introduced and validated a reproducible, end-to-end pipeline for *Mining of Reusable Component Libraries through Unsupervised Multi-Modal Clustering*. Starting from 128,883 raw meshes extracted from 202 real student IFC submissions, we built a 1,700-object benchmark across 17 merged IFC classes; composed three modalities (handcrafted geometry, VLM-derived text, visual foundation-model embeddings); designed and ablated a family of fusion recipes spanning per-block PCA and UMAP, raw and variance-balanced concatenation, UMAP-on-concatenation, late Borda rank fusion and canonical-correlation projection; ran ≈29,400 clustering configurations across the full hyperparameter grid; and produced two deployable library applications.

The headline result is a multi-modal clustering at **macro purity** 0.853 **(bootstrap mean** 0.848 ± 0.008**), bootstrap ARI** 0.874 ± 0.019**,** $k = 126$ **clusters**, on the 1,700-object benchmark of 17 merged IFC classes. The winning configuration is F5_varbal: the three modality blocks are variance-balanced, concatenated, and projected to four dimensions with UMAP, then clustered with HDBSCAN at `method=leaf`, `min_cluster_size=5`, `min_samples=3` and force-assignment. A supervised Random-Forest trained on the F5_varbal winning representation reaches macro recall 0.857, an indicative upper bound (a different metric from cluster purity) that the unsupervised macro purity 0.853 essentially meets; a richer per-modality PCA-8 representation lifts that ceiling to 0.891. Late-Borda retrieval on the same three modalities reaches **recall@10** $=$ **0.966**.

Substantively, the work establishes three findings worth highlighting:

- **Targeted handcrafted features remain competitive.** Four engineered geometric features (orientation pc_1^z and pc_3^z; cross-section aspect ratio; normal entropy; horizontal-face fraction) raise the OBB baseline from macro 0.434 to 0.704 at $k = 32$, each targeting an explicit cluster-confusion pattern. Geometry alone is competitive with the full multi-modal fusion on 10 of 17 IFC types.

- **Prompt engineering is the dominant lever for text.** A seven-version progression (v1 generic geometric tags through v7 IFC-aware) lifts text-only macro from 0.534 to 0.740, a larger gain than swapping the embedding model. The crucial reducer

choice (UMAP > PCA, universal across versions) was identified only after a 10-reducer ablation.

- **Fusion architecture matters more than encoder choice.** Among visual encoders the top three are within < 1 pp. Among fusion recipes, the best (F5_varbal) and the worst (plain F5) differ by 11.7 pp despite being the same UMAP-on-concatenation projection: the only difference is whether the modality blocks are variance-balanced first. *Variance-balancing each block before the global projection* dominates every alternative we tested, including the per-block PCA and CCA-corrected designs.

The k-matched K-Means versus HDBSCAN comparison clarifies a methodological point that the first-pass narrative had obscured: HDBSCAN's contribution at the fusion winner is not partition quality at a given k but *automatic granularity discovery*. At matched $k = 126$, K-Means reaches macro purity 0.863 ± 0.006 and HDBSCAN-force 0.853: the headline number is therefore a property of the F5_varbal representation at the data-natural granularity, with HDBSCAN correctly identifying $k \approx 126$ from the data (the schema-grain $k = 17$ understates the natural library granularity by a factor of seven) and HDBSCAN-force additionally yielding a complete no-noise partition suitable for the deployed catalog.

Finally, the library-mining angle of the work is not merely metric-driven: the auto-generated catalog and the recall@10=0.966 retrieval both give an architect or library curator concrete, navigable artefacts. The catalog evidence (Chapter 5, Section 5.12.1) shows that the recovered sub-clusters correspond to architecturally meaningful component families: six distinct staircase sub-types, eight distinct furniture sub-types and three distinct wall sub-types surface from a flat 17-class IFC schema. The full *reusable* claim, in the sense of cross-project generalisable component families, is operationally pending the library-utility evaluation described in Section 7.2; what this thesis has shown is that the latent component structure exists and can be mined from the meshes alone, without the need for consistent authoring, manual tagging, or label supervision.

7.2 Future Work

The midterm presentation closed with a six-item future-work list; the priorities for the defence and beyond are as follows.

7.2.1 Cross-corpus evaluation: IFCNet (done) and BIMGEOM (open)

The most pressing midterm-time follow-up was **cross-corpus generalisation**. This has now been completed for IFCNetCore [16]; the full audit is in Chapter 6. The 9

handcrafted geometry features, applied to IFCNet without retuning, reach a supervised F1-macro of 0.857 on 20 native IFC classes, confirming they are not over-fitted to the TUM distribution; F1a / F1a_umap is the universal supervised winner and F5_varbal the universal unsupervised winner across both corpora; and the trimodal recipe ranking generalises across datasets with Spearman $\rho \approx 0.93$.

What remains open is (i) the **BIMGEOM** industrial-scale corpus and (ii) a real-world federated repository in a tougher modeller distribution. On IFCNet the Gemma 4 text did not lift trimodal past the bimodal geometry-plus-vision result.

7.2.2 Library-utility evaluation

The thesis title commits to "reusable component libraries." To *earn* the word *reusable* we need to demonstrate that the 126 mined clusters generalise across the 202 source submissions:

- **Cross-submission overlap.** For each cluster, compute the fraction of source submissions whose objects appear in it. A cluster whose members come from many submissions represents a true cross-project reusable pattern; a cluster concentrated in one submission is a project-specific artefact.
- **Within-cluster redundancy quantification.** A measurement of geometric redundancy inside each cluster (for example chamfer-distance histograms between cluster members) would quantify how much of the mined library could be consolidated into a smaller set of canonical components.

7.2.3 Out-of-schema retrieval

A natural-language retrieval evaluation with 15–20 hand-written queries (e.g. "find me thin curtain-wall mullions", "find me operable double-leaf doors") against the 1,700-object catalog would validate the open-vocabulary use case. Recall@10 on this out-of-schema queries set is the metric.

7.2.4 Agentic retrieval via MCP4IFC

A second deployment vector is to expose the fused-embedding index as a tool to an LLM agent via the MCP [59]. **MCP4IFC** [60], an LLM-driven generative building-design framework, already pairs an MCP server with the `ifcopenshell` APIs; adding a "search the component library by natural-language query" tool that calls our F4 late-Borda fusion would extend such an agent from authoring elements to retrieving from a mined library, closing the loop from agent thought to library hit.

7.2.5 Deep learning frontier

The thesis sits firmly in the "handcrafted features + foundation models" design space. A natural extension is to train a small per-element encoder end-to-end, either a PointNet++ on the vertex cloud or a multi-view CNN on the renders, fine-tuned with a contrastive objective using the 126 HDBSCAN clusters as pseudo-labels. The supervised RF ceiling on a richer PCA-8 representation (0.891 macro recall) sits a few points above the unsupervised winner; a learned per-element representation may help close that representational gap.

7.2.6 Long-term: integration into BIM authoring tools

The most impactful long-term direction is integrating the catalog into the BIM authoring loop itself: a Revit / OpenBuildings plug-in that, when a user is about to create a new family, surfaces the nearest existing library entries via the F4 retrieval index, and flags whether the existing entry can be reused rather than duplicated. This closes the loop from research artefact to production tool.

7.3 Closing

The thesis title "**Mining of Reusable Component Libraries through Unsupervised Multi-Modal Clustering**" commits to four ideas: (i) mining, in the sense of unsupervised discovery; (ii) reusable, in the sense of cross-project generalisable; (iii) component libraries, in the sense of architecturally-meaningful groupings; and (iv) unsupervised multi-modal clustering, in the sense of using more than one signal source without label supervision. Each is partially earned by the work in this document and each has an explicit follow-up identified above. The pipeline, benchmark, and downstream applications are reproducible from the open-source code in the repository; the headline numbers (macro 0.853, recall@10 0.966, a 126-cluster catalog) are stable to bootstrap resampling.

The latent reusable-component structure exists in BIM data, and this thesis showed that it can be mined from the meshes alone. Earning the remaining piece of the title — demonstrating that the mined components are *reusable* across submissions and projects — is the next step.

8 Supplementary Material

This appendix collects material that supports but does not belong in the main narrative: the full sweep tables, a reproducibility checklist, and the raw per-type purity tables.

8.1 Full Sweep Counts

Table 8.1 breaks down the total number of clustering runs persisted to disk across every sweep.

Table 8.1: Total clustering runs persisted to disk.

Sweep	**Rows**
Geometry single-modality	174
Text single-modality	8,568
Visual single-modality	4,284
Fusion	16,200
Late fusion (retrieval)	180
Supervised ceiling	70
Total clustering runs	**∼29,400**

Plus 70 Random-Forest 5-fold CV runs and 45 bootstrap ARI pairs.

8.2 The 48-Approach Comparison Matrix

A consolidated leaderboard combines every modality × reducer × algorithm tuple. The best-per-modality summary already appeared in Chapter 5, Table 5.7; the full 48-row matrix is too wide for the body of the thesis but is available in the repository.

8.3 HDBSCAN Hyperparameter Grid

The full grid is given in Table 8.2. Across all four single-modality sweeps and three fusion subsets, the winning HDBSCAN configuration is consistently `leaf`, `mcs`=5, `ms`=3,

Table 8.2: The 30-combination HDBSCAN grid.

`min_cluster_size` (`mcs`)	$\{5, 10, 15, 20, 30\}$
`min_samples` (`ms`)	$\{3, 5, 10\}$
`cluster_selection_method`	$\{\texttt{eom}, \texttt{leaf}\}$
Total combinations	$5 \times 3 \times 2 = 30$

`force=True`. Excess-of-mass (`eom`) is the close second; `force=False` consistently loses on macro purity (it preserves the noise label).

8.4 Reproducibility Checklist

The full pipeline reproduces from a clean checkout in a few hours on an Apple-Silicon laptop (M-series, 16 GB RAM), running the single-modality sweeps, the fusion sweep, and a downstream-regeneration step that chains the supervised ceiling, late fusion, the stability experiments, and the mining-artefact builders.

The winner configuration is data-driven: the global-best HDBSCAN row is selected from the full fusion sweep, and if a re-run changes the winner, the downstream steps follow automatically.

8.5 Per-Type Purity at the Fusion Winner

Table 8.3 lists the per-type purity of the winning fusion partition (F5_varbal geo+text+visual, HDBSCAN at $k = 126$), sorted from best to worst. These seventeen numbers are the per-class breakdown that averages to the headline macro purity of 0.853. The structural and visually distinctive types (Stair, Footing, Railing, MEP, CurtainWall, Furniture) sit at or above 0.91, while the identity-ambiguous tail (Door, Member, Plate, Window) stays in the 0.71–0.76 band; these are the same hard classes discussed throughout Chapter 5.

8.6 Per-Type Retrieval Recall@10

Table 8.4 gives the per-type breakdown of the late-Borda retrieval (Chapter 5, Section 5.13). A type's recall@10 is the fraction of its objects for which at least one object of the same type appears among the ten nearest neighbours returned by the fused ranking. Every type clears 0.90, and five reach a perfect 1.00; the lowest are Plate and Window (0.91–0.92), the same geometry-ambiguous classes that are hardest to cluster. The per-type values average to the headline macro recall@10 of 0.966.

Table 8.3: Per-type purity at the F5_varbal geo+text+visual / HDBSCAN fusion winner ($k = 126$, 1,700-object benchmark). Macro = 0.853.

Type	Per-type purity
Stair	0.96
Footing	0.96
Railing	0.94
MEP	0.93
CurtainWall	0.93
Furniture	0.91
LightFixture	0.89
Column	0.87
Covering	0.87
Wall	0.86
Beam	0.86
Slab	0.80
Roof	0.78
Window	0.76
Plate	0.74
Member	0.73
Door	0.71
Macro	**0.853**

8.7 Final Headline Numbers

For ease of reference:

- Feature engineering: macro purity 0.434 $\rightarrow$ 0.704 at $k = 32$, +0.197 (4-feature OBB $\rightarrow$ 9 features).
- Text prompt engineering: macro purity 0.534 $\rightarrow$ 0.740 across v1 $\rightarrow$ v7 (UMAP-8 reducer, HDBSCAN).
- Visual SigLIP coloured / UMAP-16: macro 0.779.
- Best unsupervised (F5_varbal geo+text+visual): macro purity at $k = 126$ is **0.853** with HDBSCAN-force (leaf, mcs=5, ms=3; bootstrap 0.848 ± 0.008) and $\mathbf{0.863 \pm 0.006}$ with K-Means at the same k. The $k = 126$ is discovered by HDBSCAN from the data; HDBSCAN-force is the deployed choice because it gives

Table 8.4: Per-type recall@10 of the late-Borda `geo+text+visual` retrieval on the 1,700-object benchmark.

Type	**recall@10**	**Type**	**recall@10**
Stair	1.00	Beam	0.99
Furniture	1.00	Column	0.99
CurtainWall	1.00	Slab	0.98
Wall	0.99	Roof	0.96
Footing	0.99	Member	0.94
MEP	0.99	Door	0.93
Railing	1.00	Window	0.92
LightFixture	0.91	Plate	0.91
Covering	1.00		
Macro recall@10			**0.966**

a complete, no-noise partition. HDBSCAN-no-force reaches 0.885 on assigned objects but leaves 20.1% as noise.

- Supervised ceiling (RF 5-fold CV): macro recall **0.857** on the F5_varbal winning representation, 0.891 on a richer per-modality PCA-8 representation; an indicative upper bound (a different metric from cluster purity).
- Cluster stability (bootstrap ARI on winner): $\mathbf{0.874 \pm 0.019}$.
- Late-fusion retrieval (Borda over three modalities) recall@10: **0.966**.

These are the numbers the rest of the thesis was about.

Abbreviations

BIM Building Information Modeling

IFC Industry Foundation Classes

AEC Architecture, Engineering and Construction

OBB Oriented Bounding Box

AABB Axis-Aligned Bounding Box

PCA Principal Component Analysis

UMAP Uniform Manifold Approximation and Projection

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

GMM Gaussian Mixture Model

EM Expectation-Maximization

RF Random Forest

ARI Adjusted Rand Index

VLM Vision-Language Model

ViT Vision Transformer

CLIP Contrastive Language-Image Pre-training

MCP Model Context Protocol

MEP Mechanical, Electrical and Plumbing

CCA Canonical Correlation Analysis

List of Figures

List of Tables

Bibliography

[1] L. McInnes, J. Healy, and J. Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." In: *arXiv preprint arXiv:1802.03426* (2018).

[2] J. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations." In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. 1967, pp. 281–297.

[3] R. J. G. B. Campello, D. Moulavi, and J. Sander. "Density-Based Clustering Based on Hierarchical Density Estimates." In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Springer, 2013, pp. 160–172.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. "Emerging Properties in Self-Supervised Vision Transformers." In: *ICCV*. 2021.

[5] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, et al. "DINOv3." In: *arXiv preprint arXiv:2508.10104* (2025).

[6] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. "Vision Transformers Need Registers." In: *ICLR*. 2024.

[7] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. "Sigmoid Loss for Language Image Pre-Training." In: *ICCV*. 2023.

[8] A. Radford, J. W. Kim, et al. "Learning Transferable Visual Models from Natural Language Supervision." In: *ICML*. 2021.

[9] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, et al. "SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features." In: *arXiv preprint arXiv:2502.14786* (2025).

[10] H.-H. Lee, Y. Zhang, and A. X. Chang. "DuoDuo CLIP: Efficient 3D Understanding with Multi-View Images." In: *arXiv preprint arXiv:2406.11579* (2024).

[11] F. Noardo, K. Arroyo Ohori, T. Krijnen, and J. Stoter. "An Inspection of IFC Models from Practice." In: *Applied Sciences* 11.5 (2021), p. 2232.

[12] B. Koo and B. Shin. "Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models." In: *Journal of Computational Design and Engineering* 5.4 (2018), pp. 391–400.

[13] A. Borrmann, M. König, C. Koch, and J. Beetz. *Building Information Modeling: Technology Foundations and Industry Practice*. Springer, 2018.

[14] T. H. Beach, Y. Rezgui, H. Li, and T. Kasim. "A rule-based semantic approach for automated regulatory compliance in the construction sector." In: *Expert Systems with Applications* (2015).

[15] N. Chen, X. Lin, H. Jiang, and Y. An. "Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology." In: *Buildings* 14.7 (2024), p. 1983.

[16] C. Emunds, N. Pauen, V. Richter, J. Frisch, and C. van Treeck. "IFCNet: A Benchmark Dataset for IFC Entity Classification." In: *arXiv:2106.09712* (2021).

[17] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. "Multi-View Convolutional Neural Networks for 3D Shape Recognition." In: *ICCV*. 2015.

[18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. "Dynamic Graph CNN for Learning on Point Clouds." In: *ACM Transactions on Graphics* (2019).

[19] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao. "MeshNet: Mesh Neural Network for 3D Shape Representation." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019.

[20] C. Emunds, N. Pauen, V. E. Richter, J. Frisch, and C. van Treeck. "SpaRSE-BIM: Classification of IFC-based geometry via sparse convolutional neural networks." In: *Advanced Engineering Informatics* 53 (2022), p. 101641. DOI: 10.1016/j.aei.2022.101641.

[21] M. Yang, X. Hei, H. Meng, K. Chen, X. Tong, Y. Li, and Q. Zhao. "Lightweight framework for IFC element classification using multi-view and heterogeneous graph with decision-level fusion." In: *Automation in Construction* (2026).

[22] J. Wu and J. Zhang. "New Automated BIM Object Classification Method to Support BIM Interoperability." In: *Journal of Computing in Civil Engineering* 33.5 (2019), p. 04019033. DOI: 10.1061/(ASCE)CP.1943-5487.0000858.

[23] B. Koo, S. La, N.-W. Cho, and Y. Yu. "Using support vector machines to classify building elements for checking the semantic integrity of building information models." In: *Automation in Construction* 98 (2019), pp. 183–194.

[24] B. Koo, R. Jung, and Y. Yu. "Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks." In: *Advanced Engineering Informatics* 47 (2021), p. 101200. DOI: 10.1016/j.aei.2020.101200.

[25] Z. Xu, Z. Xie, X. Wang, and M. Niu. "Automatic Classification and Coding of Prefabricated Components Using IFC and the Random Forest Algorithm." In: *Buildings* 12.5 (2022), p. 688.

[26] S. Tang, T. Bito, and K. Shide. "Automatic Classification of BIM Object Based on IFC Data Using the Uniclass Classification Standard." In: *Buildings* 15.13 (2025), p. 2347. DOI: 10.3390/buildings15132347.

[27] J. Abualdenien and A. Borrmann. "Ensemble-learning approach for the classification of Levels Of Geometry (LOG) of building elements." In: *Advanced Engineering Informatics* 51 (2022), p. 101497. DOI: 10.1016/j.aei.2021.101497.

[28] F. C. Collins, A. Braun, M. Ringsquandl, D. M. Hall, and A. Borrmann. "Assessing IFC Classes with Means of Geometric Deep Learning on Different Graph Encodings." In: *Proceedings of the 2021 European Conference on Computing in Construction (EC3)*. Rhodes, Greece, 2021.

[29] F. C. Collins, M. Ringsquandl, A. Braun, D. M. Hall, and A. Borrmann. "Shape encoding for semantic healing of design models and knowledge transfer to scan-to-BIM." In: *Proceedings of the Institution of Civil Engineers – Smart Infrastructure and Construction* 175.4 (2022).

[30] H. Luo, G. Gao, H. Huang, Z. Ke, C. Peng, and M. Gu. "A Geometric-Relational Deep Learning Framework for BIM Object Classification." In: *Computer Vision – ECCV 2022 Workshops*. Lecture Notes in Computer Science. Springer, 2023. DOI: 10.1007/978-3-031-25082-8_23.

[31] Y. Yu, H. Lee, W. Lee, and B. Koo. "Transformer-based multi-view learning for BIM clash classification: Penetration taxonomy and constructability analysis." In: *Journal of Computational Design and Engineering* 13.4 (2026), pp. 227–251.

[32] C. Jin, M. Xu, L. Lin, and X. Zhou. "Exploring BIM Data by Graph-based Unsupervised Learning." In: *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*. 2018, pp. 582–589.

[33] Z. Wang and H. Ying. "Graph Representation Learning: Embedding Multimodality BIM Models into Graphs." In: *Proceedings of the 2025 European Conference on Computing in Construction (EC3)*. 2025.

[34] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. "Shape Distributions." In: *ACM Transactions on Graphics* (2002).

[35] J. Demantké, C. Mallet, N. David, and B. Vallet. "Dimensionality based scale selection in 3D LiDAR point clouds." In: *ISPRS Workshop Laser Scanning*. 2011.

[36] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015), pp. 286–304.

[37] C. T. Zahn and R. Z. Roskies. "Fourier Descriptors for Plane Closed Curves." In: *IEEE Transactions on Computers* (1972).

[38] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors." In: *Symposium on Geometry Processing* (2003).

[39] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. "On Visual Similarity Based 3D Model Retrieval." In: *Computer Graphics Forum* (2003).

[40] J. W. H. Tangelder and R. C. Veltkamp. "A Survey of Content Based 3D Shape Retrieval Methods." In: *Multimedia Tools and Applications* (2008).

[41] R. B. Rusu, N. Blodow, and M. Beetz. "Fast Point Feature Histograms (FPFH) for 3D Registration." In: *ICRA* (2009).

[42] F. Tombari, S. Salti, and L. Di Stefano. "Unique Signatures of Histograms for Local Surface Description." In: *ECCV*. 2010.

[43] A. E. Johnson and M. Hebert. "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes." In: *IEEE PAMI* (1999).

[44] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." In: *CVPR*. 2017.

[45] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space." In: *NeurIPS*. 2017.

[46] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun. "Point Transformer." In: *ICCV*. 2021.

[47] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu. "PCT: Point Cloud Transformer." In: *Computational Visual Media* (2021).

[48] M. Oquab, T. Darcet, et al. "DINOv2: Learning Robust Visual Features without Supervision." In: *TMLR* (2024).

[49] M. Liu et al. "OpenShape: Scaling Up 3D Shape Representation Towards Open-World Understanding." In: *NeurIPS* (2023).

[50] L. Xue et al. "ULIP-2: Towards Scalable Multimodal Pre-training for 3D Understanding." In: *CVPR* (2024).

[51] H. Liu, C. Li, Q. Wu, and Y. J. Lee. "Visual Instruction Tuning." In: *NeurIPS*. 2023.

[52] J. Bai et al. "Qwen-VL: A Versatile Vision-Language Model." In: *arXiv preprint arXiv:2308.12966* (2023).

[53] G. Gemini Team. "Gemini: A Family of Highly Capable Multimodal Models." In: *arXiv preprint arXiv:2312.11805* (2024).

[54] T. Luo et al. "Cap3D: Scalable 3D Captioning." In: *NeurIPS* (2023).

[55] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. "Multimodal Machine Learning: A Survey and Taxonomy." In: *IEEE PAMI* (2018).

[56] K. Gadzicki, R. Khamsehashari, and C. Zetzsche. "Early vs Late Fusion in Multimodal Convolutional Neural Networks." In: *Fusion*. 2020.

[57] H. Hotelling. "Relations Between Two Sets of Variates." In: *Biometrika* (1936).

[58] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. "Deep Canonical Correlation Analysis." In: *ICML*. 2013.

[59] Anthropic. *Introducing the Model Context Protocol*. `https://www.anthropic.com/news/model-context-protocol`. 2024.

[60] B. K. Nithyanantham, T. Sesterhenn, A. Nedungadi, S. Peral Garijo, J. Zenkner, C. Bartelt, and S. Lüdtke. "MCP4IFC: IFC-Based Building Design Using Large Language Models." In: *arXiv preprint arXiv:2511.05533* (2025).