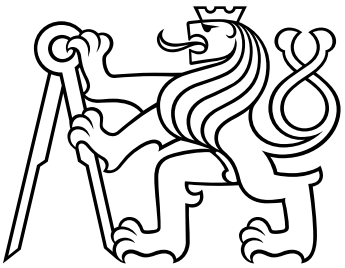
Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Physics



Master's Thesis

Self-supervised feature extraction from break junction traces for enhanced clustering

Oliver Klimt

klimtoli@fel.cvut.cz
https://icluto.oklimt.com

Supervisor: Ing. Ladislav Sieger CSc.
Supervisor specialist: RNDr. Jaroslav Vacek, Ph.D., RNDr. Jindřich Nejedlý, Ph.D.

Study programme: Cybernetics and Robotics

May 2026

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Klimt Oliver** Personal ID number: **499151**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Self-supervised feature extraction from break junction traces for enhanced clustering

Master’s thesis title in Czech:

Extrakce příznaků z křivek experimentu break-junction metodou self-supervised learningu pro následné klastrování

Name and workplace of master’s thesis supervisor:

Ing. Ladislav Sieger, CSc. Department of Physics FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **23.01.2026** Deadline for master's thesis submission: **22.05.2026**

Assignment valid until: **by the end of summer semester 2026/2027**

Head of department’s signature

Vice-dean´s signature on behalf of the Dean

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work.
The student must produce his thesis without the assistance of others, with the exception of provided consultations.
Within the master’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Bc. Klimt Oliver

Student’s signature

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Klimt Oliver** Personal ID number: **499151**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Self-supervised feature extraction from break junction traces for enhanced clustering

Master’s thesis title in Czech:

Extrakce příznaků z křivek experimentu break-junction metodou self-supervised learningu pro následné klastrování

Guidelines:

1) Research machine learning architectures for auto-encoders, transformers and learned descriptors.
2) Research applicable loss functions.
3) Propose an encoder that maps traces into a latent/descriptor space.
4) Based on the latent space cluster traces with K-Means with large K (e.g. K=30) and compare the clusters with previous clustering runs on histogram features for the same K.

Bibliography / sources:

1] BENAVIDES-CESAR, Llinet; MANSO-CALLEJO, Miguel-Ángel; CIRA, Calimanut-Ionut. Methodology Based on BERT (Bidirectional Encoder Representations from Transformers) to Improve Solar Irradiance Prediction of Deep Learning Models Trained with Time Series of Spatiotemporal Meteorological Information. Forecasting, 2025, 7.1: 5.
2] SHIEH, Jin; KEOGH, Eamonn. i SAX: disk-aware mining and indexing of massive time series datasets. Data Mining and Knowledge Discovery, 2009, 19.1: 24-57.
3] PAN, Zhichao, et al. Data-Driven Insights in Single-Molecule Break Junction Studies: A Comprehensive Review of the Data Analysis Methods. Langmuir, 2025, 41.36: 24152-24174
4] MUTSCHLER, Christopher, et al. (ed.). Unlocking Artificial Intelligence: From Theory to Applications. Springer, 2024.
5] LIN, Luchun, et al. Spectral clustering to analyze the hidden events in single-molecule break junctions. The Journal of Physical Chemistry C, 2021, 125.6: 3623-3630.

FAKULTA ELEKTROTECHNICKÁ
FACULTY OF ELECTRICAL ENGINEERING
Technická 2
166 27 Praha 6

DECLARATION

I, the undersigned

Student's surname, given name(s):	Klimt Oliver
Personal number:	499151
Programme name:	Cybernetics and Robotics

hereby declare that the Master's Thesis titled

Self-supervised feature extraction from break junction traces for enhanced clustering

is my own independent work and that I have declared all sources of information used in accordance with the Methodological Guideline for adhering to ethical principles when elaborating an academic final thesis and the Framework Rules for the use of artificial intelligence at CTU for study and pedagogical purposes in Bachelor and continuing Masters studies.

I declare that I have used artificial intelligence tools during the preparation and writing of the final thesis.
I've verified the generated content. I confirm that I am aware that I am fully responsible for the content of the final thesis.

In Prague on 16.05.2026

Bc. Oliver Klimt

.....
student's signature

Abstract

Break Junction experiments generate large datasets of stochastic conductance traces in which molecular junction events are rare and difficult to isolate. Existing approaches either discard trace-level information through histogram aggregation, rely on expert-tuned conductance thresholds, or require costly annotated training data.

This thesis adapts DINO, a self-supervised vision transformer, to one-dimensional conductance traces. The model learns patch-level embeddings from unlabelled data and requires no conductance-range calibration. A cosine-similarity search in the resulting embedding space cleanly separates bulk, molecular, and tunnelling-current regimes, enabling annotation-free retrieval of molecular candidates at 15.8 ms per trace on a standard laptop. Cross-instrument validation on the University of Copenhagen bp4k dataset reaches $\mathrm{F1} \approx 0.78$ without retraining, indicating that the representation is instrument-invariant.

Keywords: break junction, single-molecule conductance, self-supervised learning, DINO, vision transformer, representation learning, molecular electronics, clustering

Abstrakt (CZ)

Experimenty typu *break junction* generují rozsáhlé datové soubory stochastických vodivostních křivek, v nichž jsou události molekulárních spojů vzácné a obtížně izolovatelné. Stávající přístupy buď zahazují informaci na úrovni jednotlivých křivek agregací do histogramů, spoléhají na expertně laděné prahy vodivosti, nebo vyžadují nákladná anotovaná trénovací data.

Tato práce adaptuje model DINO, *self-supervised vision transformer*, na jednorozměrné vodivostní křivky. Model se učí *embeddingy* na úrovni jednotlivých *patchů* z neoznačených dat a nevyžaduje kalibraci rozsahu vodivosti. Vyhledávání pomocí kosinové podobnosti ve výsledném prostoru *embeddingů* čistě odděluje režimy objemové vodivosti, molekulárního spoje a tunelovacího proudu, což umožňuje vyhledávání molekulárních kandidátů bez nutnosti anotace s rychlostí 15,8 ms na jednu křivku na běžném notebooku. Validace napříč přístroji na datovém souboru bp4k z Kodaňské univerzity dosahuje $\mathrm{F1} \approx 0{,}78$ bez nutnosti přetrénování, což naznačuje, že je naučená reprezentace nezávislá na konkrétním přístroji.

Klíčová slova: break junction, vodivost jednotlivých molekul, self-supervised learning, DINO, vision transformer, učení reprezentací, molekulární elektronika, shlukování

Acknowledgement

I would like to express my sincere gratitude to Dr. Starý for the opportunity to become a member of his research group at IOCB Prague. I am deeply indebted to Dr. Nejedlý for providing the molecular data and for his invaluable mentorship regarding data interpretation.

My thanks also go to Dr. Sieger for his steadfast guidance and mentorship, and to Dr. Vacek for his expert consultation on the clustering workflow and insightful feature suggestions. Finally, I acknowledge the High Performance Computing group (HPCg) at IOCB Prague for providing the computational resources and cluster access that enabled the training of many large neural networks.

Contents

Chapter 1

Introduction

1.1 Motivation

The Mechanically Controllable Break Junction (MCBJ) experiment is a foundational technique for measuring the electrical conductance of single molecules. By providing insight into molecular charge transport, this method is essential for the advancement of molecular electronics—a field aiming to utilise single molecules as active electronic components. In contrast to the "top-down" approach of conventional semiconductor manufacturing, which relies on the miniaturisation of bulk silicon, molecular electronics follows a "bottom-up" paradigm. By building functional circuits from the molecular level up, this approach offers a pathway to bypass the physical and economic barriers encountered as traditional devices reach the atomic scale. While organic electronics are already ubiquitous in technologies such as Organic Light-Emitting Diodes (OLEDs), the transition to unimolecular devices requires understanding quantum phenomena at the nanometre scale.

The theoretical framework for molecular electronics was first proposed by Aviram and Ratner [1], who conceptualized a molecular rectifier. Today, this field relies on the synthesis of novel, complex architectures, such as those developed by the Ivo Starý research group at the Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences in Prague. To evaluate these candidates for organic electronics, the break junction setup measures electrical conductance as a function of electrode displacement. This process generates massive datasets characterised by stochastic quantum events, necessitating advanced computational pipelines for accurate analysis.

The analysis of Break Junction data has undergone a significant methodological evolution. Traditional approaches rely on one- and two-dimensional conductance histograms [2], [3], which aggregate thousands of individual traces to reveal the most probable conductance values of a molecular junction. While effective for identifying dominant transport channels, these methods discard trace-level information and struggle to resolve rare or transient molecular configurations—a manifestation of the "curse of dimensionality" in high-dimensional feature spaces. Per-trace analysis has further benefited from algorithmic approaches adapted from other disciplines, such as econometric change-point detection for automated plateau identification [4].

To address the remaining challenge of high-dimensional feature spaces, more recent work has incorporated machine learning, using Principal Component Analysis (PCA) to compress the feature space [5] and supervised Convolutional Neural Networks (CNNs) to classify individual traces [6]. However, supervised approaches require large, expertly labelled training sets, which are costly to produce in experimental settings. This has motivated a shift towards self-supervised learning (SSL) methods [7], which can extract meaningful representations directly from unlabelled data—a more practical paradigm given the scale of modern MCBJ datasets.

1.2 Research Goals

The computational pipeline developed in our prior work [8] paired supervised snap-back detection with dimensionality reduction: traces were filtered, encoded as conductance histograms, reduced by PCA, and partitioned by unsupervised K-means or DBSCAN. This approach distinguished molecular from blank tunnelling traces and could resolve coarse differences in molecule–electrode coupling strength, but failed to resolve rare stochastic events — a structural limitation analysed in detail in Chapter 2.

To address the limitations of linear dimensionality reduction, we transitioned from Principal Component Analysis (PCA) to a learned feature representation. By employing a deep learning approach—specifically designed to capture non-linear dependencies—we aim to preserve the high-dimensional variance associated with rare, stochastic events. Learning the feature set directly from the raw data prevents the collapse of these unique signatures into an unrecoverable latent space, enabling the detection and classification of individual molecular events that the previous pipeline had obscured.

A further limitation of the previous pipeline was its sensitivity to the specified conductance range. Figure 1.1 shows the two-dimensional conductance histogram of the validation dataset.[1] When the range was carefully tuned to 10^{-4}–10^{-1} G/G_0, the clusters cleanly separated molecular from blank traces (Figure 1.2). With the default range of 10^{-6} –10^{-1} G/G_0, however, clustering degraded: both clusters were dominated by the limit region rather than by distinct molecular structure (Figure 1.3). This sensitivity reflects the method's lack of intrinsic molecular awareness — differentiation relied entirely on expert-chosen conductance bounds rather than on features learned from the data.

1.3 Thesis Structure

- **Chapter 1** motivates the need for improved MCBJ data analysis, reviews the limitations of linear dimensionality reduction in our prior work, and states the research goals of this thesis.
- **Chapter 2** provides the theoretical background of the MCBJ experiment and the state-of-the-art in molecular conductance data analysis.
- **Chapter 3** discusses the principles of Self-Supervised Learning and the architectural choices for the DINO-based feature extraction.
- **Chapter 4** details the development of the computational framework, including the 1D Vision Transformer backbone and the data processing pipeline.
- **Chapter 5** presents the analysis of the learned representations, comparing the proposed method with previous clustering approaches.
- **Chapter 6** summarizes the findings and outlines potential directions for future research in automated trace classification.

[1] The colormap of the 2D histogram is dominated by the high density of data points in the snap-back region, which saturates the scale and renders lower-density features in other conductance regions less visible.

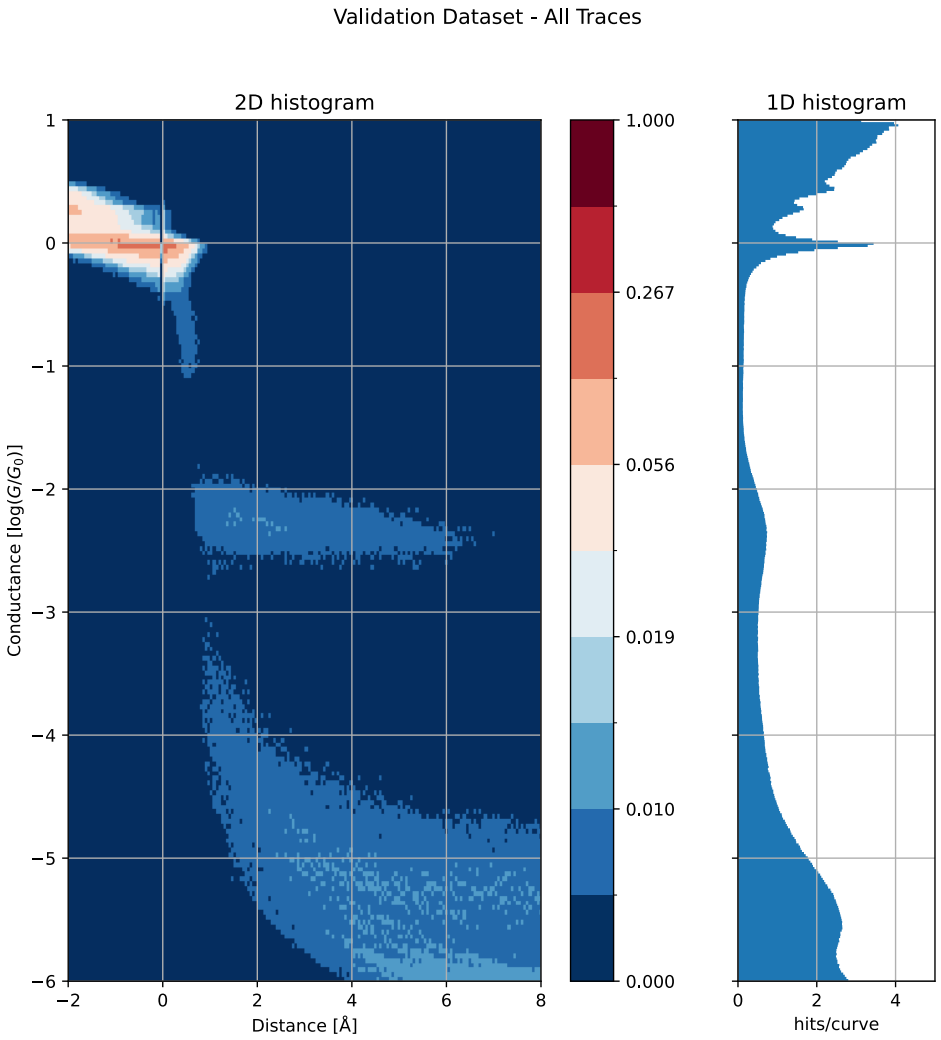
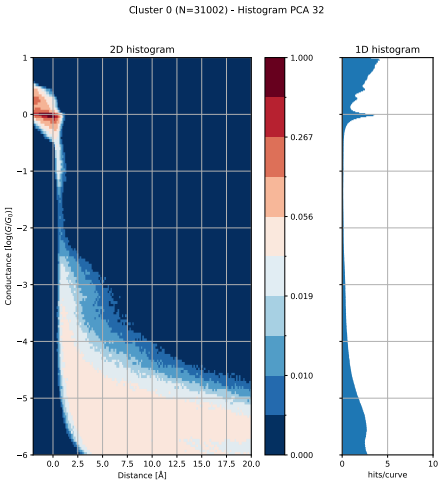
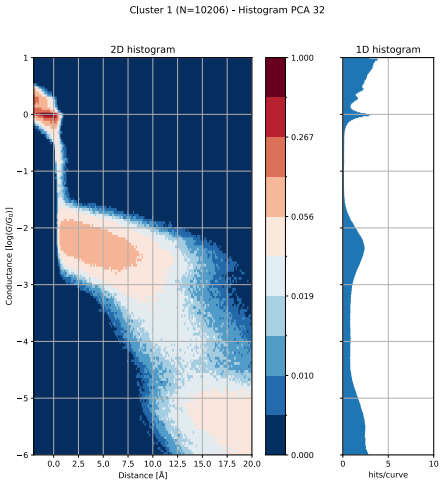


Figure 1.1: Two-dimensional histogram of the validation dataset, showing the distribution of conductance traces prior to the clustering analysis.

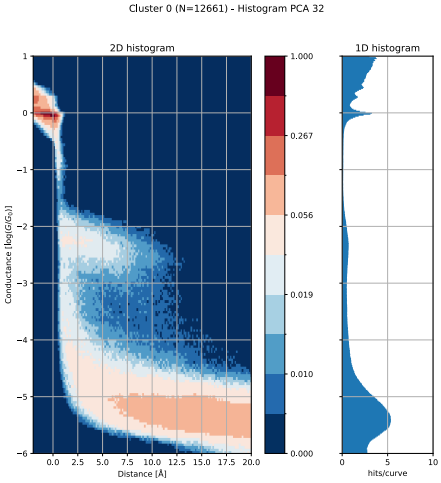


(a) Cluster A - no molecule

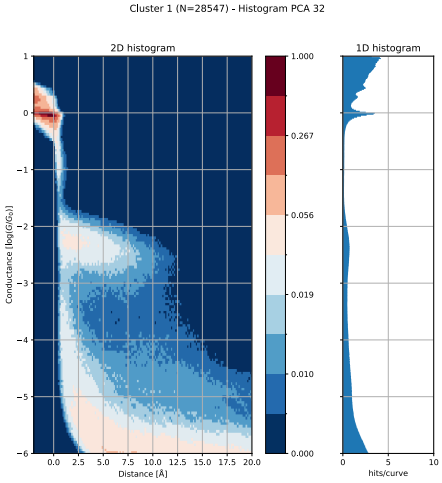


(b) Cluster B - molecule present

Figure 1.2: Example of clusters obtained in our prior work using linear PCA and K-means clustering for *hist32* feature that was computed for the range of 10^{-4} to 10^{-1} G/G_0.



(a) Cluster A



(b) Cluster B

Figure 1.3: These two clusters were obtained using the same method as in Figure 1.2, but with the default range of 10^{-6} to 10^{-1} G/G_0.

Chapter 2

Theoretical Background

This chapter establishes the experimental and conceptual foundations for the analysis developed in this thesis. It covers the physics of single-molecule conductance, the MCBJ measurement technique, and the computer vision paradigm that motivates the trace representation strategy used in subsequent chapters.

2.1 Single-Molecule Electronics

In traditional silicon-based electronics, the continuous miniaturization of transistors to pack more computational power onto a single chip is rapidly approaching fundamental physical limits, where silicon can no longer function reliably. Molecular electronics offers an alternative paradigm to overcome these boundaries: instead of carving progressively smaller features out of bulk materials in a top-down approach, circuits are constructed from the bottom-up, using individual molecules as the fundamental building blocks [9].

Within this field, single molecules can be engineered to perform distinct electronic functions. For instance, a molecule can act as a:

- **Wire:** Conducting electricity efficiently between two defined points.
- **Transistor:** Switching current flow on and off in response to an external gate voltage.
- **Diode:** Permitting current to flow in only one preferred direction.

Leveraging these molecular-scale functionalities, researchers have successfully demonstrated a diverse range of single-molecule devices, including photodiodes, rectifiers, light-emitting diodes, molecular switches and molecular wires [10], [11].

2.2 Traces, Conductance and G_0

The investigation of single-molecule electronics relies on the precise characterisation of charge transport at the atomic scale. In a typical experiment, a metallic junction—most commonly gold—is repeatedly formed and ruptured using break junction techniques (described in detail in Section 2.3). As the contact narrows to a single atom, the electrical conductance becomes quantized, exhibiting discrete plateaus at integer multiples of the quantum of conductance, G_0 [12], defined by the relation:

$$G_0 = 2\frac{e^2}{h} \tag{2.1}$$

where e is the elementary charge and h is the Planck constant. This value represents the maximum conductance of a single, perfectly transmitting quantum channel, serving as a fundamental calibration point for the experiment.

Upon further retraction, the metallic contact ruptures, allowing a single molecule to bridge the gap. By collecting thousands of these conductance-displacement traces and compiling them into histograms, researchers can statistically determine the most probable molecular conductance [13]. Collecting a sufficiently large number of traces is essential for this statistical interpretation to be meaningful, as discussed in the tutorial by York

et al. [14]. This normalised approach, which reports values as fractions of G_0, allows for a consistent analysis of the electronic transparency of a range of molecular structures regardless of the specific experimental setup [15].

A typical trace is depicted in Figure 2.1 and is formed by the following steps:

1. **Contact formation** — the electrodes are pushed together, establishing a metallic junction.
2. **Retraction** — the electrodes are pulled apart, thinning the metallic contact until it ruptures.
3. **Molecular bridging** — following the snap-back, a molecule bridges the electrode gap.
4. **Repetition** — as the electrodes are stretched further, the metal–molecule–metal bond breaks and the cycle repeats.

Each trace is segmented into the following regions and characteristic events:

1. **Bulk** — a high-conductance plateau corresponding to metallic contact between the electrodes.
2. **N-atom bridge** — a conductance plateau corresponding to a contact formed by N parallel atomic chains (typically $N = 1, 2, 3$).
3. **Snap-back** — a sudden drop in conductance marking the rupture of the last atomic bridge (i.e. the 1-atom bridge).
4. **Molecular bridge** — a low-conductance plateau where a single molecule bridges the electrode gap.
5. **Tunnelling** — conductance decays exponentially as the electrode separation widens.
6. **Limit** — conductance reaches the noise floor of the measurement setup (typically 10^{-6} –10^{-8} G/G_0).

These regions describe the structure of a single pulling-apart cycle. Because the fixed 2500-datapoint measurement window can extend slightly past the cycle, a reconnection event — an abrupt return to the bulk regime as the electrodes are pulled back together — is occasionally captured at the tail of a trace.

2.3 Break Junction Measurement Setup

Break junction experiments have been realised in two principal configurations. In the Scanning Tunneling Microscope Break Junction (STM-BJ) approach [13], a sharp metallic tip is repeatedly driven into and retracted from a flat substrate; the vertical displacement is controlled by a piezoelectric element, allowing thousands of junctions to be formed and broken in rapid succession. The Mechanically Controllable Break Junction (MCBJ) technique [12] instead mounts a notched metallic wire on a flexible substrate; bending the substrate displaces the wire's two halves at the sub-ångström level, providing exceptional mechanical stability and low noise. In both configurations, gold (Au) is the dominant electrode material, owing to its chemical inertness and strong affinity for thiol anchor groups. Alternative metals such as copper (Cu) have also been explored, as the choice of electrode influences the energy-level alignment between molecule and contact [16]. The apparatus used in this work supports both MCBJ and STM-BJ configurations with gold electrodes, and is described in the following subsections.

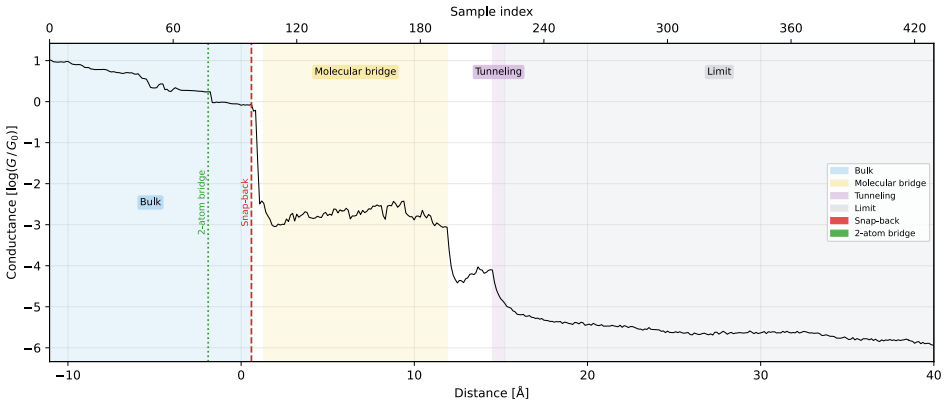
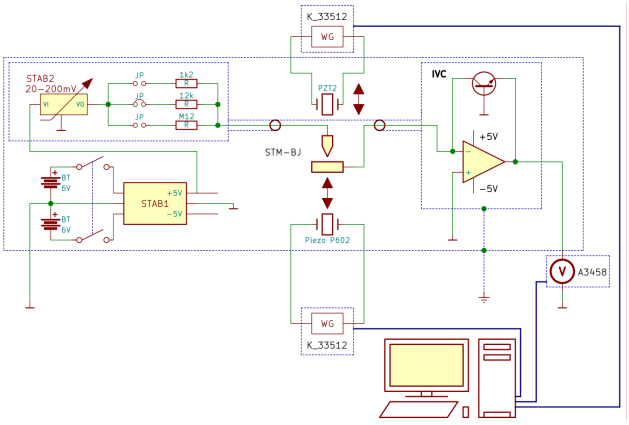


Figure 2.1: An annotated example of a single conductance-displacement trace (zoomed in to the snap-back and post-snap region). The characteristic regions are labelled: the high-conductance bulk plateau, the 2-atom bridge, the snap-back event marking the rupture of the metallic contact, the molecular bridge plateau, and the tunnelling regime where conductance decays exponentially with electrode separation.

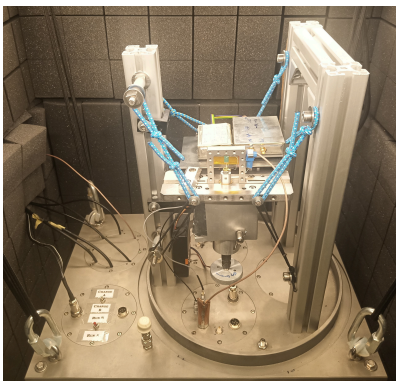
2.3.1 Experimental Apparatus

2.3.1.1 Design and Specifications

Since commercially viable options for high-precision break junction measurements are currently unavailable, the research utilises a custom-built apparatus (Figure 2.2). This system was designed and constructed in collaboration with Prof. Josef Zicha (CTU) and Jiří Miletín (BMD, s.r.o Teplice). The hardware configuration at the Institute of Organic Chemistry and Biochemistry (IOCB) has remained unchanged since 2024, and the same apparatus has also been utilised in the work of Nejedlý [17]. The device is engineered for extreme sensitivity, capable of detecting electrical currents as low as 100 fA with a sampling rate reaching 200 kSa/s. This high resolution is necessary to capture the discrete conductance steps as the metal wire thins to a single-atom contact and eventually breaks.

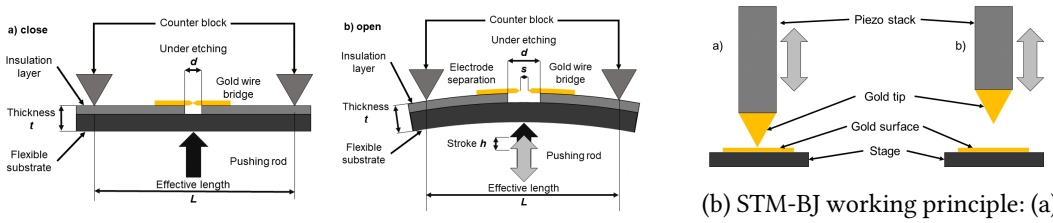


(a) Circuit schematic of the custom-built apparatus. Taken from [17].



(b) Photograph of the custom-built measurement apparatus installed at IOCB. Photo: Dr. J. Nejedlý.

Figure 2.2: Custom-built STM-BJ apparatus used in this work.



(a) MCBJ working principle: (a) closed and (b) open junction.

(b) STM-BJ working principle: (a) closed and (b) open junction.

Figure 2.3: Working principles of the MCBJ (left) and STM-BJ (right) configurations. Taken from [17].

2.3.1.2 Mechanical Configuration and Operation

The apparatus is highly versatile, supporting both Mechanically Controllable Break Junction (MCBJ) and Scanning Tunneling Microscopy-Based Break Junction (STM-BJ) configurations. The working principles of both configurations are depicted in Figure 2.3. In the STM-BJ mode, which provided the majority of the datasets for this study, the electrodes are aligned vertically: a single pointy gold tip moves relative to a gold substrate. The mechanical separation is driven by a pushing rod activated by a piezo crystal capable of a 100 N pushing force. This crystal is controlled via a Keysight K_33512B wave generator, allowing for precise, repeatable electrode displacement.

2.3.1.3 Signal Processing and Data Acquisition

To ensure signal integrity and manage the wide dynamic range of conductance, several custom electronic components are integrated:

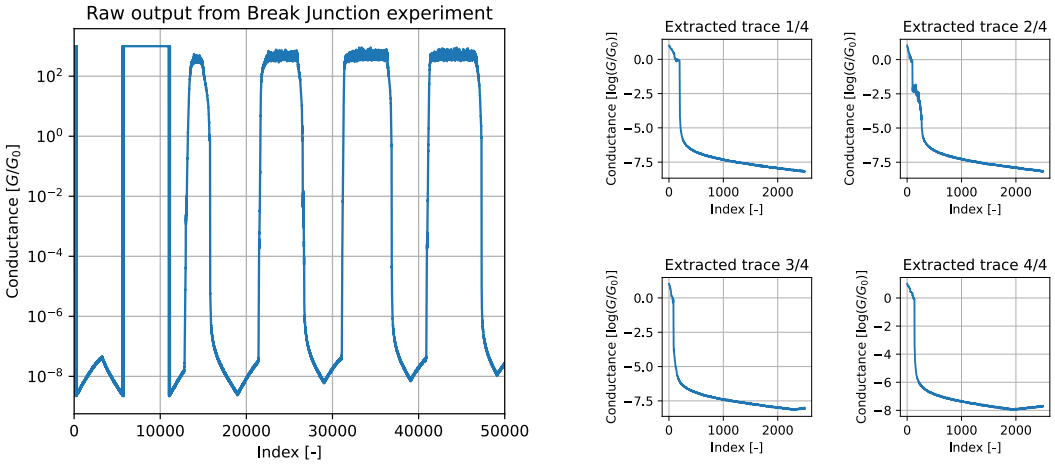
- **Voltage Stabilisation:** Two custom stabilisers (STAB1 and STAB2) are used; one stabilises power from lead-acid batteries, while the second is shielded to offset measurement bias.
- **Current-to-Voltage Conversion:** A custom-made converter (IVC) utilises a logarithmic operational amplifier to cover several orders of magnitude in conductance within a single trace.
- **Measurement:** The resulting output voltage is sensed by an Agilent 3458 digital multimeter.
- **Software:** Data acquisition is managed through a LabView-based environment.

To minimise external interference, all sensitive electronic parts are strictly shielded and grounded.

2.3.1.4 Trace Extraction

During a typical measurement session, the electrode separation cycle is repeated thousands of times: the piezo crystal drives the electrodes together and then apart in a continuous sawtooth pattern, producing one conductance-displacement trace per cycle. All measured voltage samples are written continuously to a set of plain-text (`.txt`) files, where each file contains a long, concatenated record of raw conductance values from many successive cycles, as shown in the left panel of Figure 2.4a.

To recover individual traces from this raw stream, a two-step segmentation procedure is applied. First, a learned snap-back detector combined with Non-Maximum Suppression (NMS) locates the characteristic abrupt conductance drop that marks the rupture of the



(a) Raw input (b) Extracted traces

Figure 2.4: Preprocessing workflow: raw conductance data as recorded by the STM-BJ setup, and four traces extracted from the raw data for further processing.

last metallic atomic bridge in each cycle. Second, the onset of each trace is anchored at the point where the log-conductance crosses $\log(G/G_0) = 1$, typically a few hundred data points before the detected snap-back. The result is a fixed-length segment containing the bulk plateau, the atomic chain regime, the snap-back event, and the subsequent molecular bridge or tunnelling region, as shown in the right panel of Figure 2.4.

2.4 Inspiration from Computer Vision

The design of our analysis pipeline draws heavily on a paradigm shift that occurred in computer vision over the past two decades: from describing whole signals globally to describing them *locally*, segment by segment. This perspective proved transformative in image analysis and motivates the approach we adapt here for molecular conductance traces.

2.4.1 Local Descriptors and Bag of Visual Words

A foundational insight in computer vision is that images are better understood through their *local structure* than through pixel-level global statistics. The Scale-Invariant Feature Transform (SIFT) [18] formalized this idea: rather than representing an image as a single high-dimensional vector, SIFT detects repeatable keypoints—corners, blobs, edges—and produces a compact descriptor for each local patch surrounding them. These descriptors are invariant to scale, rotation, and illumination changes, making them highly transferable across scenes.

The Bag of Visual Words (BoVW) model [19] leverages such local descriptors to represent entire images in a compact yet expressive form. The key steps are:

1. **Feature extraction:** Extract local descriptors from keypoints across a large collection of images.
2. **Codebook construction:** Cluster all extracted descriptors (e.g., via k-means) into K cluster centroids, forming a "visual vocabulary."

3. **Encoding:** Map each image to a histogram of how frequently its descriptors fall into each visual word.
4. **Retrieval and clustering:** Compare images by comparing their histograms.

The power of this approach is that it reduces a complex, high-dimensional signal into a sparse, interpretable representation—one that captures *which patterns are present*, rather than where exactly they occur. Crucially, rare patterns receive their own visual words, making the representation inherently capable of preserving infrequent events that global methods average away.

2.4.2 Structure from Motion and Descriptor Databases

The scalability of local descriptors became most apparent in Structure from Motion (SfM) [20], where thousands of photographs of a scene are matched and assembled into a 3D reconstruction purely by querying a shared database of descriptors. Each image contributes its local features to a shared vocabulary; correspondences are established not by comparing full images, but by finding matching visual words. This demonstrates that a descriptor database built from local patches can serve as a universal retrieval index—a concept directly relevant to our trace similarity search task.

2.4.3 Segmentation as a Prerequisite: SAM

A complementary development is the Segment Anything Model (SAM) [21], which demonstrated that a neural network can learn to segment arbitrary objects in images in a zero-shot fashion, without task-specific supervision. SAM's key insight is that *segmentation is a generally learnable skill*: given enough diverse visual data, a model can discover where meaningful boundaries lie. In our pipeline, this justifies using a learned model—specifically, a 1D U-Net—to locate and extract the tunnelling region of each trace before further processing.

2.4.4 Learned Descriptors: DINO

While hand-crafted descriptors like SIFT are powerful, their expressiveness is ultimately bounded by what a human designer can formalize. Self-supervised learning approaches have shown that neural networks can learn far richer local representations directly from data. DINO [22], [23], [24] is one such approach: it trains a vision transformer to produce patch-level feature embeddings via a self-distillation objective, requiring no labels. A key empirical finding is that DINO's patch tokens exhibit strong semantic coherence—spatially adjacent, visually similar regions cluster tightly in embedding space. This makes DINO features ideal building blocks for a BoVW-style pipeline, replacing hand-crafted keypoint descriptors with deeply learned, domain-adaptive ones.

2.4.5 The Analogy to Conductance Traces

The parallel to our problem is direct. A conductance trace is a 1D signal composed of recurring, physically meaningful segments: the bulk plateau, N-atom bridges, snap-back, molecular bridge, and tunnelling current. Much like an image contains corners, edges, and textured regions, a trace contains these morphological elements in varying combinations, durations, and conductance levels. No two traces are identical, yet the underlying vocabulary of segments is shared across thousands of measurements.

If we can describe individual segments of a trace with a compact, learned descriptor—as DINO does for image patches—we can:

1. Build a **visual vocabulary** of trace segments from a large unlabelled dataset.
2. **Encode** each trace as a histogram over this vocabulary.
3. **Cluster and retrieve** traces based on their BoVW representation, naturally preserving rare molecular events rather than absorbing them into dominant global components.

This is the central pipeline developed in the following chapters. The DINO architecture (Chapter 3) provides the learned segment descriptor; the BoVW construction and subsequent clustering are detailed in the implementation chapter (Chapter 4).

2.5 Data Analysis Challenges

2.5.1 The "Curse of Dimensionality" and the "Uniform Effect"

In our prior work we used PCA for dimensionality reduction. This approach assumes a linear relationship between the data points, which is not always the case in single-molecule conductance traces. Traces are visualised in logarithmic scale, which is a non-linear transformation, but in logarithmic scale an exponential decay appears as a linear trend. Dimensionality reduction is therefore necessary to obtain a compact representation on which clustering algorithms can be applied in finite time.

The problem with PCA is that it is very limited in capturing rare events in traces, these events are often lost in the latent space, which makes it impossible to cluster them. Consequently, principal component analysis allowed us to overcome the curse of dimensionality, but it introduced a new problem, the loss of rare events. This loss of rare events is often called the uniform effect. The mechanism is rooted in PCA's objective. PCA identifies the d orthogonal directions in $\mathbb{R}^D$ that maximise total projected variance across the dataset. When $N_r \ll N$ traces carry a rare event and the remaining $N_c = N - N_r$ do not, the sample covariance matrix is weighted accordingly:

$$C = \frac{N_c}{N} C_{\text{common}} + \frac{N_r}{N} C_{\text{rare}} \approx C_{\text{common}} \tag{2.2}$$

The leading eigenvectors of C therefore reflect the dominant variance structure of the common traces. A rare event—such as a molecular conductance plateau at a specific G/G_0 level—may manifest along a direction v^* that contributes to C with weight $N_r/N \ll 1$. Unless the rare-event variance in direction v^* greatly exceeds the background, v^* will not appear among the retained components. Upon projection, rare-event traces collapse onto the same latent locus as surrounding common traces, making their distinct physical structure invisible to any downstream clustering algorithm. While PCA has been applied to break junction data to extract configuration-specific information [5], its global variance objective fundamentally limits its ability to resolve minority populations.

The pipeline employed in the preceding work [8] proceeded in four stages: raw traces were filtered to remove artefacts and incomplete junctions; fixed-length feature vectors were extracted from the tunnelling region (conductance histograms); PCA reduced their dimensionality to a tractable subspace; and K-means clustering was applied in that latent space. The pipeline is illustrated in Figure 2.5.

Figure 2.5: The four-stage analysis pipeline from our prior work: raw traces are filtered, encoded as conductance histograms, reduced by PCA, and finally partitioned by K-means.

Non-linear dimensionality reduction methods such as t-SNE [25] and UMAP [26] can better separate minority clusters than PCA and were therefore considered as a replacement for the PCA step. However, the core problem is not the reduction method itself—it is the input: the conductance histogram summarizes an entire trace into a single global vector, discarding all local shape information. Swapping PCA for a non-linear method does not fix this. This work therefore takes a different approach: instead of reducing a global histogram, it builds a learned descriptor directly from local trace segments, as described in the following chapters.

Chapter 3

Self-Supervised Learning

The central methodology of this research relies on replacing traditional, heuristic-based data processing with a foundational deep learning model. Specifically, we employ a Self-Supervised Learning (SSL) approach using the DINO (Self-Distillation with No Labels) architecture. DINO acts as the foundational tool for our entire analytical pipeline, enabling us to extract rich, unbiased feature representations directly from the raw molecular junction traces. By learning the underlying morphology of the conductance data without human intervention, DINO provides the mathematical basis for all subsequent clustering and similarity search tasks discussed in this thesis.

3.1 SSL Concepts

To overcome the limitations of supervised snap-back detection, we employ a framework that learns feature representations without any human-labelled data. The motivation for avoiding manual annotation is twofold.

First, the sheer volume of data generated by MCBJ experiments—often hundreds of thousands of traces—makes comprehensive labelling practically impossible. While our initial supervised models were severely limited by the size of a manually annotated subset, an SSL approach scales effortlessly to the full dataset.

Second, molecular conductance traces are inherently stochastic, exhibiting complex, non-linear quantum phenomena. Human annotators rely on macroscopic visual heuristics, introducing subjective bias and inter-annotator inconsistency. Self-supervised learning sidesteps this by forcing the network to uncover the intrinsic mathematical structure of the data directly, potentially revealing subtle molecular events that heuristic inspection would miss.

Self-supervised learning is a broad family of methods that construct a supervisory signal directly from unlabelled data. The defining principle is the *pretext task*: a proxy objective whose solution forces the network to build a useful internal representation of the input. Three families of pretext tasks are relevant here.

Reconstruction-based methods train the network to reproduce its own input. The simplest form is the autoencoder (Figure 3.1), which compresses the trace into a low-dimensional latent code and reconstructs it exactly. A more powerful variant, the masked autoencoder (MAE) [27], withholds a random portion of the input and requires the network to predict the missing content, compelling the encoder to capture longer-range structure.

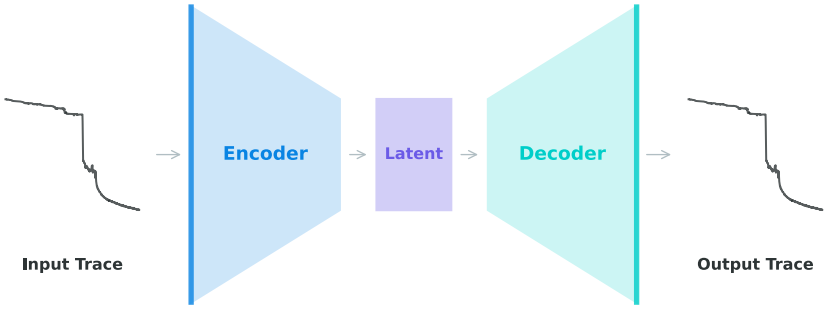


Figure 3.1: Conceptual diagram of an autoencoder architecture.

Contrastive learning takes a different approach: the model is trained to embed different augmented views of the same sample close together while pushing apart views from different samples [28].

Self-distillation methods, to which DINO belongs, replace explicit positive/negative pairs with a second, slowly evolving copy of the network that generates the supervisory signal.

The choice of pretext task directly shapes what the learned representation captures. Reconstruction-based methods must retain enough information to reproduce the entire input. For our traces this is counterproductive: the bulk contact and limit regions together account for roughly 80% of each trace, so a reconstruction objective is dominated by these uninformative segments and the molecular plateau—the signal of interest—contributes too little to drive the representation.

In this thesis we need to go a step further: rather than representing the whole trace, we want to identify individual segments and perform clustering or similarity search based on those segments alone. Self-distillation, and DINO in particular, is better suited to this goal.

3.2 DINO Architecture

A novel approach to self-supervised learning is called self-distillation with no labels (DINO) [22], [23], [24]. Unlike traditional architectures that rely on massive labelled datasets or contrastive pairs, DINO extracts powerful representations by training a network to predict the output of its own dynamically updated momentum encoder.

In this architecture, we utilise two parallel networks: a student network and a teacher network. They share the exact same structural architecture, but their weights are parameterised differently. The objective is to train the student network to match the representations produced by the teacher network. This process relies on a technique called multi-crop training. The student is fed heavily augmented, local "crops" of a molecular trace (focusing on fine-grained, localised structural details), while the teacher is given a global crop of the same trace (providing the broader context).

The student network is updated via standard gradient descent (backpropagation) to minimise the loss between its predictions and the teacher's output. Crucially, the teacher

network is not updated via backpropagation. Instead, its weights are updated using an Exponential Moving Average (EMA) of the student's weights over time. This momentum update rule stabilises the training process, effectively making the teacher an ensembled, smoother version of the student network across multiple training steps. Because the student is forced to match the high-quality, stable representations of the teacher from localised, distorted inputs, the network is forced to learn robust, invariant features of the molecular trace. The term *distillation* traces back to Hinton et al.'s knowledge distillation framework [29]; the "self" prefix reflects DINO's novelty of replacing the external teacher with a momentum-updated copy of the student.

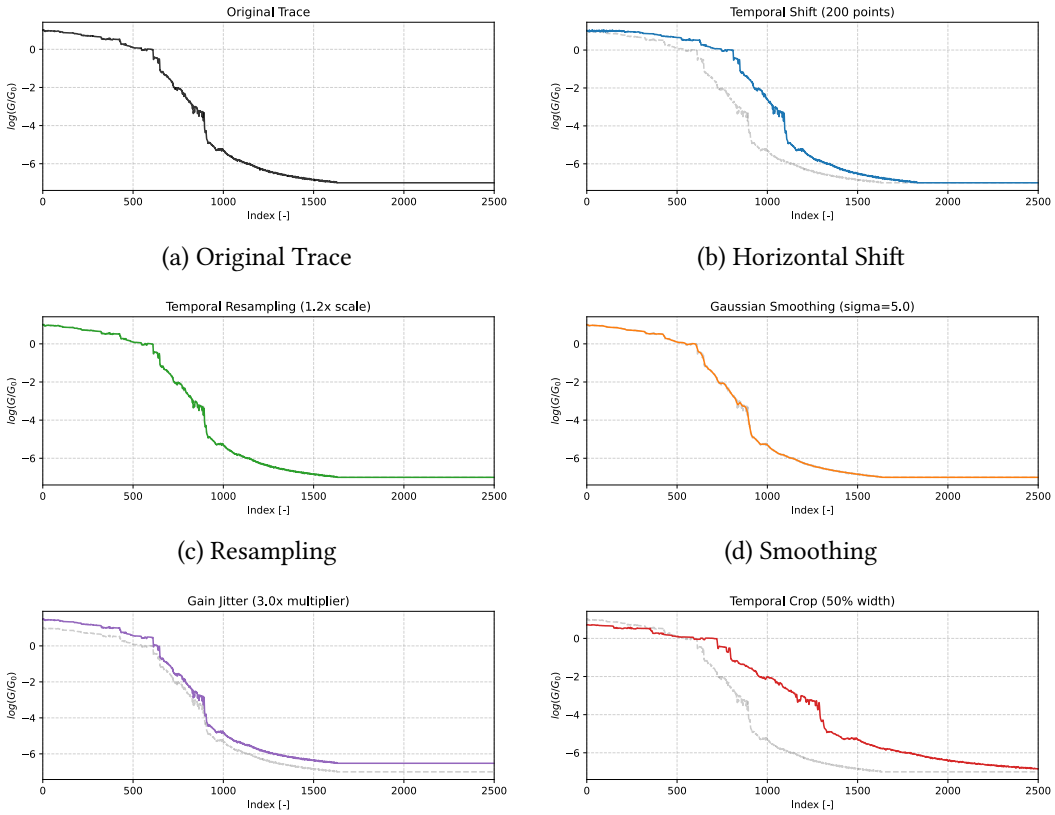
This approach has the advantage that it can be used to learn features from the data without the need for human-labelled data. Both networks must agree by embedding each view into a shared latent space. The training objective measures how well the student's output distribution matches the teacher's, using cross-entropy loss. A key risk in this setup is representational collapse, where the network finds a trivial solution by mapping every input to the same output. To prevent this, the teacher's raw predictions are re-centred before being converted to probabilities: a running average of the teacher's recent outputs is subtracted, keeping the output distribution balanced so that no single dimension dominates. In addition, the teacher uses a lower temperature parameter when converting its raw scores to probabilities. A lower temperature produces a more peaked distribution — one class gets a much higher probability than the rest — giving the student a sharper, more informative target to learn from.

A consequence of training without ground-truth labels is that DINO has no dedicated validation phase: there is no held-out metric that directly certifies the learned representations are physically meaningful. Proxy diagnostics — the training loss, or the compactness of patch clusters in embedding space (Section 4.3.1) — can be tracked to monitor trends across epochs, but final validation rests on qualitative inspection by a domain expert.

Some literature categorizes DINO as an unsupervised learning method [30], since it requires no human annotations. The more precise designation is self-supervised: the supervisory signal is not absent but is derived from the data itself. The teacher's representation serves as a dynamic pseudo-label for the student, so the network is guided by an explicit, structured objective—distinguishing it from classical unsupervised methods such as k-means, which impose no such per-sample target. Throughout this thesis we adopt the self-supervised framing.

3.2.1 Data Augmentation

Because DINO relies heavily on the student network matching the teacher network across different views of the same data, data augmentation plays a crucial role. For our 1D sequential molecular conductance traces, typical image-based augmentations (like colour jittering or rotation) are inapplicable. Instead, we generate our "local" and "global" crops by taking random contiguous subsequences of the trace. Furthermore, we apply domain-specific augmentations such as random vertical scaling (within physically plausible limits) and small amounts of Gaussian noise to simulate instrumental variance. This ensures the model learns the fundamental morphology of snap-backs and plateaus rather than memorising exact noise patterns or absolute conductance values.



(a) Original Trace (b) Horizontal Shift

(c) Resampling (d) Smoothing

(e) Vertical Gain (f) Multi-crop

Figure 3.2: The data augmentation pipeline for 1D molecular conductance traces, showing the transition from raw data to the final training samples.

The following figures illustrate the transformation steps applied during the data augmentation process. Each step is designed to make the model invariant to specific experimental variations.

The baseline trace (Figure 3.2a) represents the raw logarithmic conductance as a function of the measurement index. It contains the fundamental features we wish to capture, including the high-conductance bulk region, the characteristic molecular plateaus, and the final snap-back to the tunnelling regime.

To account for variations in the starting point of the measurement and potential jitter in the piezo actuator control, we apply a horizontal shift (Figure 3.2b). This translation ensures that the network does not rely on the absolute index position of a feature, but rather on its relative morphology within the sequence.

Temporal resampling (Figure 3.2c) is used to stretch or compress the trace along the distance axis. This augmentation simulates variations in piezo actuator speed or sampling frequency, forcing the model to recognise molecular signatures regardless of their duration or "velocity" in the raw data stream.

Gaussian smoothing (Figure 3.2d) is applied to reduce high-frequency instrumental noise. By blurring the signal slightly, the model is encouraged to focus on the robust macroscopic

structure of the trace—such as the length and slope of plateaus—rather than overfitting to microscopic noise patterns or quantisation artefacts.

Vertical gain (Figure 3.2e) scales the conductance values by a random factor. This step is critical for ensuring that the feature embeddings are invariant to absolute conductance levels, which can vary depending on the molecule-electrode coupling strength or changes in the pre-amplifier gain settings.

Finally, the multi-crop strategy (Figure 3.2f) extracts multiple overlapping global and local subsequences from the augmented trace. The global views (80–100%) preserve the overall "story" of the trace, while the local views (20–40%) zoom in on specific sub-structures. By training the student to predict the global representation from these small local snippets, the model learns to identify "visual words" (patterns) that are characteristic of the trace's identity regardless of where they appear.

Overall, these augmentations create a challenging "pretext task" where the student must understand the essence of a trace from a noisy, shifted, and heavily cropped version of it, leading to the robust, clusterable features used in our subsequent Bag-of-Visual-Words pipeline.

3.3 Data Preparation & Curation

The training dataset consists of more than 400 000 traces, which include molecules synthesised by Dr. Nejedlý, namely JIN206, JIN466, JIN467, JIN536, and JIN537. These molecules were selected because they represent newly synthesised compounds with varied molecular structures, providing the model with a broad range of conductance signatures during training. The named molecules were dissolved in mesitylene (MesH), a solvent chosen for its slow evaporation rate, which ensures a stable molecular environment throughout the measurement.

As our validation dataset we have chosen 41 000 traces of R296 molecules, synthesised by Ing. Arnošt Seidler. Thanks to their simple structure, they are a good candidate for a validation dataset and we know how much of a displacement they have (13 Å).

Because self-supervised learning relies heavily on the underlying dataset distribution, this introduces a strong data quality dependency; if the dominant features in the training set are irrelevant, the resulting representations will not faithfully encode the molecular signals of interest.

Chapter 4

Implementation

This chapter describes the implementation of the three-stage analysis pipeline. The first stage trains a DINO model on raw conductance traces to produce patch-level embeddings that capture local structural features without any manual labels. The second stage applies a Bag-of-Visual-Words (BoVW) encoding: patch embeddings are quantized into a fixed vocabulary, and each trace is represented as a frequency histogram over visual words. The third stage uses a similarity search pipeline to retrieve traces whose patch embeddings closely match a user-defined reference signature. The chapter opens with an overview of iCluto, the Python platform in which all three stages are implemented, before describing each stage in turn.

4.1 iCluto

iCluto is a command-line interface application written in Python. It builds on PyTorch for deep learning workloads and Scikit-learn for classical machine learning algorithms, providing researchers at IOCB with a unified environment for every stage of break junction data analysis.

The tool originated as a clustering application in the preceding work [8], where it implemented the four-stage pipeline described in Chapter 2: raw trace filtering, conductance histogram extraction, PCA-based dimensionality reduction, and K-means clustering. Since then, its scope has broadened into a general-purpose research platform. Its current capabilities include:

- **Trace management:** Loading, merging, and filtering `TraceCollection` objects assembled from multiple experimental runs or instruments.
- **Clustering:** Applying dimensionality reduction and clustering algorithms to group traces by their conductance signatures.
- **Visualization:** Generating two-dimensional conductance–displacement histograms and diagnostic plots.
- **Annotation:** Semi-automated and manual labelling of individual traces for downstream supervised analysis.

The present work extends iCluto with the self-supervised DINO representation pipeline and Bag-of-Visual-Words encoding described in the following sections, adding more expressive trace embeddings and improved sensitivity to rare molecular events.

4.2 Model Architecture

Both the student and teacher share the same architecture, illustrated in Figure 4.1. It consists of two components: a patch embedding layer (`PatchEmbed1D`) that converts the raw conductance trace into a sequence of local feature vectors, and a Transformer backbone (`TraceTransformer`) that processes them into context-aware representations. The output of the backbone is passed through a projection head (`DINOHead`) that maps the global trace summary to a probability distribution over a fixed set of 1024 prototype

patterns, on which the DINO loss is computed [22]. The per-patch embeddings produced by the backbone play no role in the loss, but carry the structural information used by the downstream similarity-search and segmentation analyses (Chapter 5).

The architecture and distillation process are illustrated in Figure 4.1.

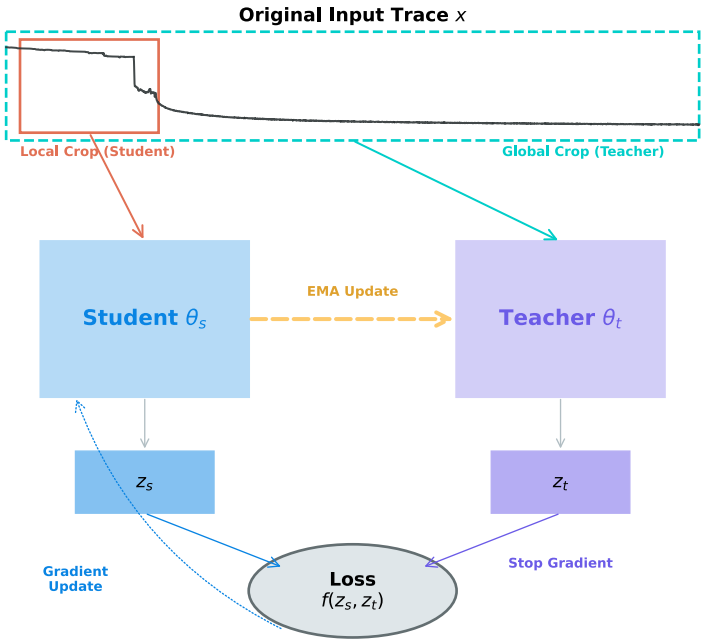


Figure 4.1: System architecture of the DINO framework for self-supervised trace representation learning. The Student network learns from local crops while being regularised by a Teacher network updated via an EMA path. z_s and z_t denote the raw output logits of the student and teacher projection heads, respectively; the DINO loss is computed by comparing their probability distributions.

To tokenise a 1D molecular conductance trace for the DINO model, the sequential data is processed through a 1D convolutional layer (`Conv1d`) that acts as a patch embedding mechanism. This convolution uses a specific kernel size corresponding to the desired patch length (e.g., 8 samples) and a stride that determines the overlap between adjacent patches (typically set to half the patch size for 50% overlap). As the convolutional kernel slides across the raw trace, it linearly projects each local window of conductance values into a higher-dimensional continuous embedding. This transforms the 1D signal into a sequence of discrete feature vectors—the "patches." These patches are then treated as individual tokens, a learnable class token (`[CLS]`) is prepended to the sequence, and 1D positional embeddings are added to preserve the temporal order before the sequence is fed into the Transformer encoder, as illustrated in Figure 4.2.

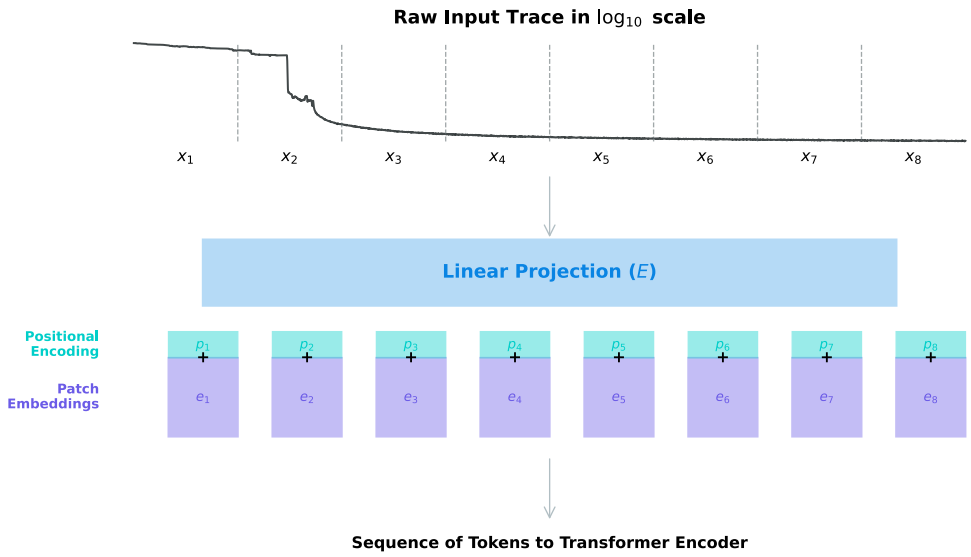


Figure 4.2: Illustration of trace segmentation into patches for feature embedding.

Unlike recurrent networks, which process tokens one at a time in sequence, a Transformer attends to all tokens simultaneously. This makes it inherently permutation-invariant: without additional information, the model would assign the same representation to the sequence $[A, B, C]$ as to $[C, A, B]$, because no positional information is encoded in the tokens themselves.

Positional embeddings solve this by adding a position-specific vector to each token's representation before it enters the attention layers. Each vector encodes the token's index in the sequence, so that two identical patches appearing at different positions in the trace receive different inputs to the Transformer. This allows the model to reason about temporal order—distinguishing, for example, whether a patch captures the noisy high-conductance bulk contact at the start of the trace, the stable low-conductance plateau of a molecular junction following the snap-back, or the exponentially decaying signal in the deep tunnelling regime at the end.

In our implementation, positional embeddings are realised as a learnable tensor: a set of vectors, one per position, that are optimised jointly with the rest of the network. They are added to the patch tokens immediately after the `[CLS]` token is prepended and before the sequence enters the Transformer layers.

Once the sequence enters the Transformer backbone, it is processed through multiple blocks containing Multi-Head Self-Attention (MHSA) and feed-forward networks. The self-attention mechanism enables the model to weigh the importance of different patches relative to one another, effectively capturing long-range structural dependencies across the trace. For example, it allows the network to correlate the length of a conductance plateau with the slope of the subsequent snap-back event.

Throughout these layers, the `[CLS]` token acts as an aggregator, accumulating contextual information from all the local patches. At the output of the final Transformer layer, the

vector corresponding to the [CLS] token serves as the comprehensive global representation of the entire measurement. It is this aggregated vector that is ultimately passed to the projection head to compute the DINO loss, forcing the student to match the teacher's holistic understanding of the trace.

The [CLS] token's role is, however, confined to training. The BoVW pipeline introduced later in this chapter, and the similarity-search and segmentation analyses of Chapter 5, operate exclusively on the per-patch embeddings at the encoder output — the [CLS] token is discarded. Self-attention nonetheless ensures that each patch embedding carries the same long-range context that shaped the [CLS] representation, so the patch tokens inherit the structural awareness that the DINO objective induced.

4.3 DINO Training & Optimisation

Training DINO requires choosing both classical hyperparameters—learning rate, batch size, number of epochs—and two parameters specific to the architecture: patch size and embedding dimension. Classical hyperparameters follow well-established scaling rules (the effective learning rate is linearly scaled with batch size as $\mathrm{LR}_{\mathrm{eff}} = \mathrm{LR}_{\mathrm{base}} \times \frac{\text{batch size}}{256}$ [31]) and are therefore less critical to tune empirically. Patch size and embedding dimension, however, depend directly on the temporal structure of the input data and the desired feature granularity, and their impact on representation quality must be evaluated experimentally.

4.3.1 Hyperparameter Analysis

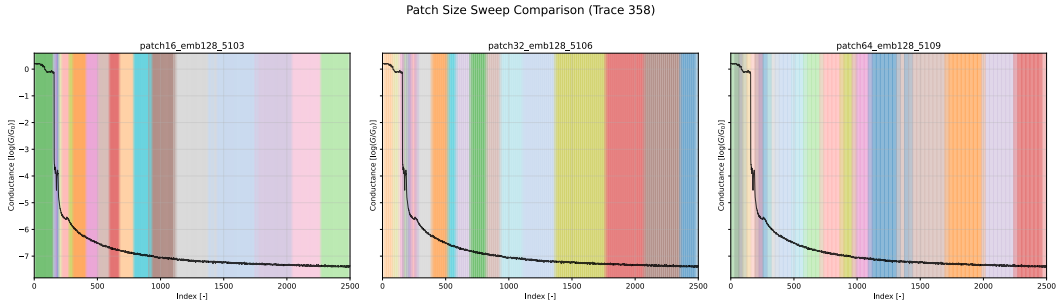
Beyond the classical training hyperparameters, the two most important architecture choices are patch size and embedding dimension. To assess their effect before committing to full training runs, we trained 15 combinations for a small number of epochs.

A smaller patch size captures finer trace detail. This is most visible in the snap-back region: comparing the columns of Figure 4.3 shows that smaller patches produce more distinct segmentation there.

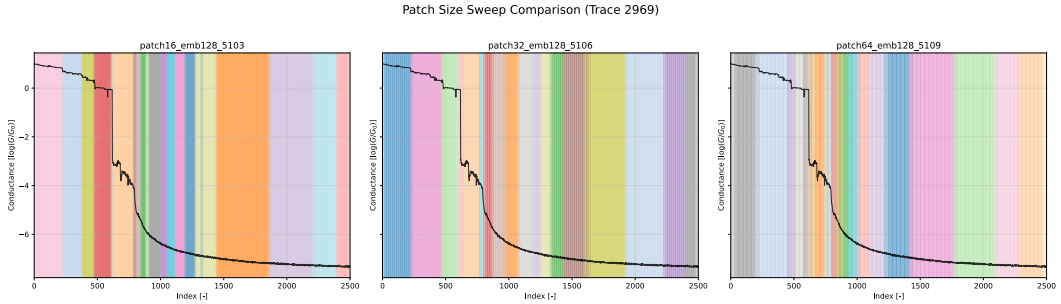
A larger embedding dimension gives the model more capacity per patch. In our short pretraining runs this advantage is not always clear—the 256-dimensional models broadly match the 128-dimensional ones (Figure 4.4). One telling difference appears at trace 2969 (Figure 4.4b): the 128-dimensional model assigns four labels to the bulk segment, while the 256-dimensional model assigns three, which better reflects the uniform character of that region.

4.3.2 Experimental Setup

All DINO training runs were performed on the High-Performance Computing Cluster (HPCC), specifically utilising the 'd' section computing nodes. These nodes are equipped with NVIDIA L40S GPUs featuring 48 GB of GDDR6 memory (CUDA architecture 89). The underlying system consists of dual AMD EPYC 9654 96-core processors running at 2.4 GHz and 363 GB of RAM, providing the necessary computational throughput for large-scale self-supervised learning on trace datasets.



(a) Trace 358



(b) Trace 2969

Figure 4.3: Evaluation of DINO performance during pretraining for different patch sizes on 2 randomly selected traces[2].

4.3.3 DINO Training

Based on the hyperparameter analysis, we trained four configurations: patch sizes 8 and 16, each with embedding dimensions of 128 and 256. The parameter counts for each are listed in Table 4.1; patch size has little effect on the total parameter count, whereas embedding dimension is the dominant factor.[3]

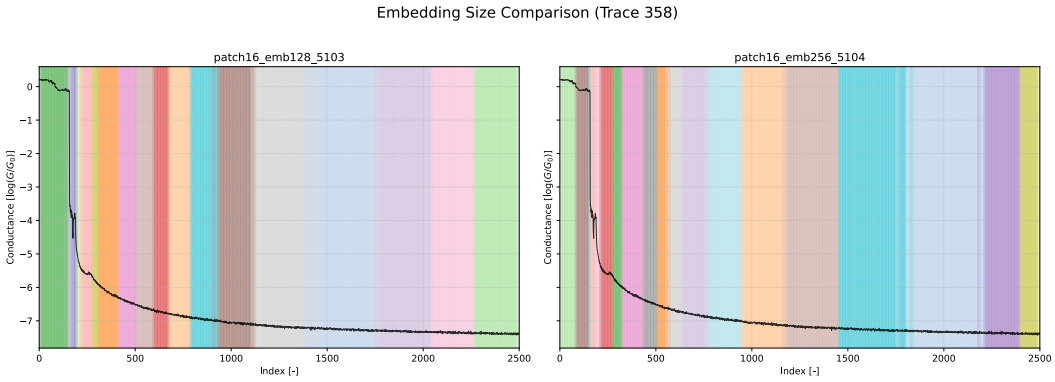
A known failure mode in self-supervised distillation is **representation collapse**: a degenerate state in which the network maps all inputs to the same constant output vector. When this occurs the student trivially satisfies the DINO objective—it always agrees with the teacher—yet the embeddings carry no discriminative information. DINO guards against this through two complementary mechanisms. **Centering** subtracts a running mean from the teacher's output, preventing any single prototype dimension from saturating and dominating the distribution. **Sharpening** via a low teacher temperature encourages peaked probability distributions that are difficult to mimic with a constant output. In our implementation an additional "Stability First" safeguard monitors the per-batch loss; if it falls below 10^{-6} for five consecutive batches—indicating imminent collapse—the teacher temperature schedule is attenuated by a factor of 0.9 to restore informative targets.

4.3.3.1 Training and Validation

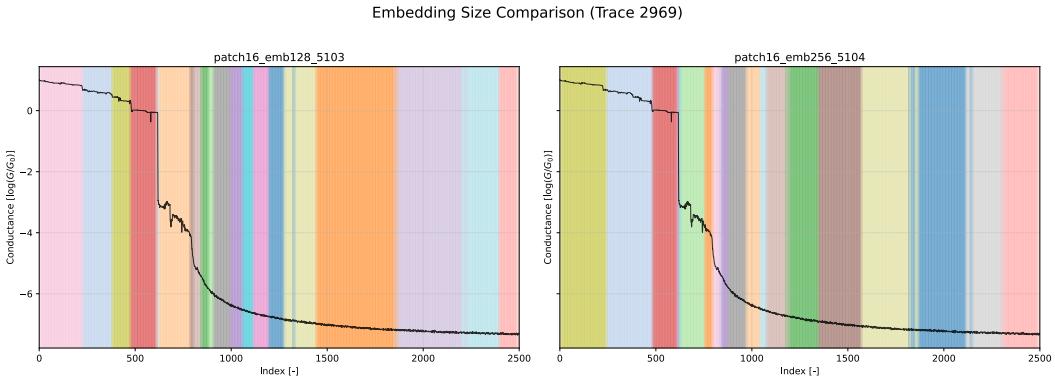
The DINO loss measures the cross-entropy between the student's and teacher's probability distributions over the prototype vocabulary; a decreasing loss indicates the student

[2]The subtitle is composed of the patch size, embedding dimension and SLURM ID.

[3]A 512-dimensional model was also considered, but the larger size would have required a significantly smaller batch size, making training prohibitively slow.



(a) Trace 358



(b) Trace 2969

Figure 4.4: Evaluation of DINO performance during pretraining for different embedding dimensions on 2 randomly selected traces.

is learning to mimic the teacher, though not necessarily that the learned representations are discriminative. The loss drops sharply within the first few epochs for all configurations (see Figure 4.5). This rapid initial convergence is explained by the scale of the training dataset: with 400 000 traces and a batch size of 32, each epoch comprises approximately 12 500 gradient update steps. By the time the loss reaches its minimum at epochs 6–19, the model has already processed between 75 000 and 240 000 weight updates—sufficient

Configuration (Patch, Dim)	**Backbone Parameters**	**Projection Head**	**Total Parameters**
Patch 16, Dim 128	835 584	722 176	1.56 Million
Patch 16, Dim 256	3 244 032	787 712	4.03 Million
Patch 8, Dim 128	874 624	722 176	1.60 Million
Patch 8, Dim 256	3 322 112	787 712	4.11 Million

Table 4.1: Parameter counts for the four DINO configurations selected for full training.

Run Configuration	Final Loss	Min Loss	Min-Loss Epoch	Max Epochs	Duration (Days)
Patch 16, Dim 128	3.8004	2.5942	19	152	4.850
Patch 16, Dim 256	3.9032	2.2992	16	82	4.847
Patch 8, Dim 128	4.1394	2.6277	16	72	4.849
Patch 8, Dim 256	3.4008	2.7912	6	39	4.848

Table 4.2: Summary of DINO training runs executed on the HPCG infrastructure.

to capture the dominant structural patterns of the trace distribution (bulk conductance, plateau, and snap-back regions). After this minimum, the loss rises substantially for all configurations. The rise is driven primarily by the teacher temperature schedule: the teacher temperature increases linearly from 0.04 to 0.07 over the first 30 epochs, producing progressively softer probability distributions that raise the cross-entropy loss even when representations continue to improve. Concurrently, the cosine learning rate schedule decays the student's update step, so the student makes smaller corrections per step and increasingly lags the teacher's slowly moving EMA targets. The patch 16, dim 256 model exhibits the sharpest rise: having achieved the lowest minimum loss of all four runs (2.2992), its distributions were the most peaked, making the subsequent temperature-driven softening disproportionately large in loss terms.

Since the DINO loss is confounded by the temperature schedule, it cannot serve as a reliable proxy for representation quality. Instead, we evaluated models using a K-means inertia metric (K=20) computed on patch embeddings extracted by the teacher network at regular intervals (see Figure 4.6). A large drop in inertia indicates that patch embeddings are organising into increasingly compact and well-separated clusters—precisely the structure required for an effective BoVW vocabulary. The 256-dimensional models exhibited a far more pronounced inertia decrease than their 128-dimensional counterparts: the patch 8, dim 256 model dropped from approximately 375 000 to 110 000 within the first 25 epochs, whereas both 128-dimensional models remained comparatively flat throughout training, suggesting that the lower embedding dimension lacks the representational capacity to form a structured latent space. Based on the largest cumulative inertia drop, we selected the patch 8, dim 256 model at epoch 30 as the backbone for all subsequent BoVW and similarity search pipelines.

The cluster distribution waterfall plots (Figure 4.7) provide a complementary perspective on representation quality. All models exhibit one dominant patch class at each epoch,

but the behaviour of that dominant class differs across configurations. The patch 16, dim 256 model shows signs of representation collapse—a single class persistently dominates —contrary to the improving inertia over the same period, suggesting the model finds a compact but degenerate solution in prototype space. The patch 8, dim 256 model also shows one dominant class per epoch, but crucially the dominant class rotates across epochs, indicating that the model is actively reorganising its vocabulary rather than locking onto a fixed prototype[4].

All four runs took approximately five days and are summarised in Table 4.2.

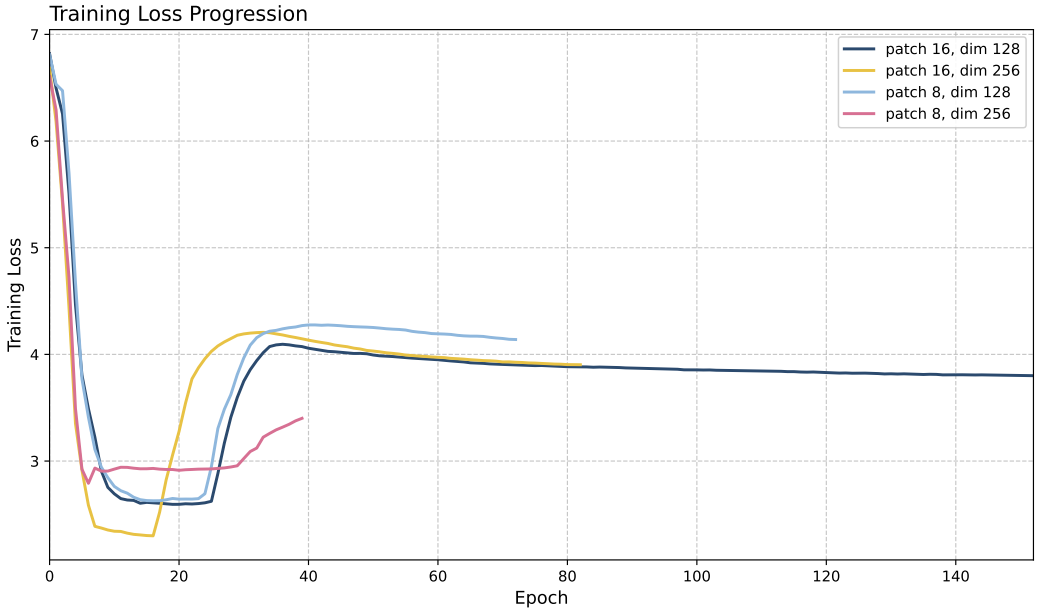


Figure 4.5: Progression of DINO training loss over epochs.

[4]K-means is non-deterministic; the colourmap is fixed for all epochs, so the assignment of visual words to colours may shift between runs. This is mitigated by rerunning K-Means several times and selecting the most consistent result.

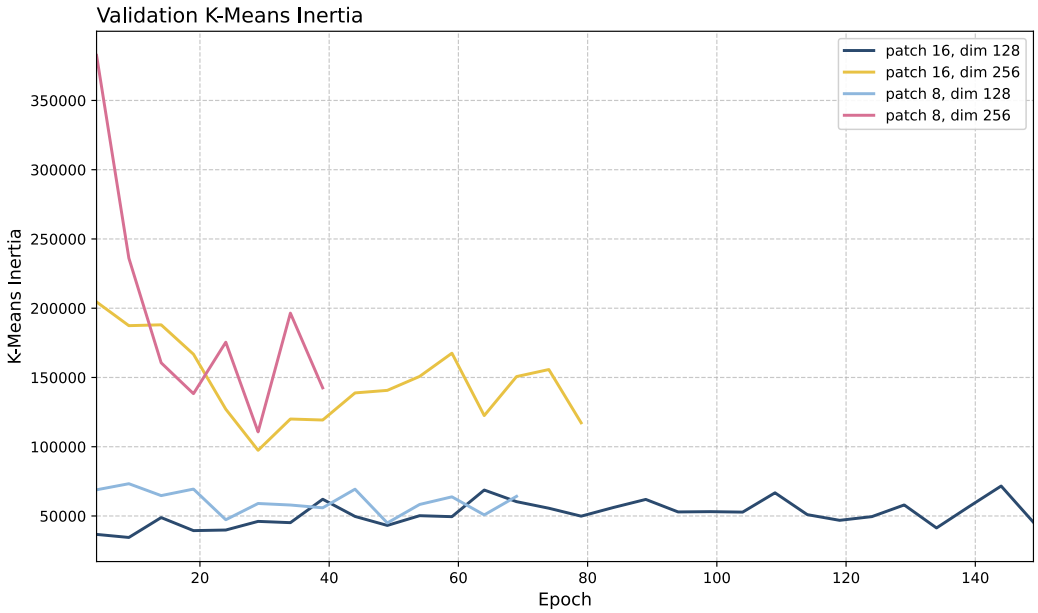


Figure 4.6: Validation K-means inertia (K=20) indicating feature cluster compactness over training.

4.4 Bag-of-Visual-Words Pipeline

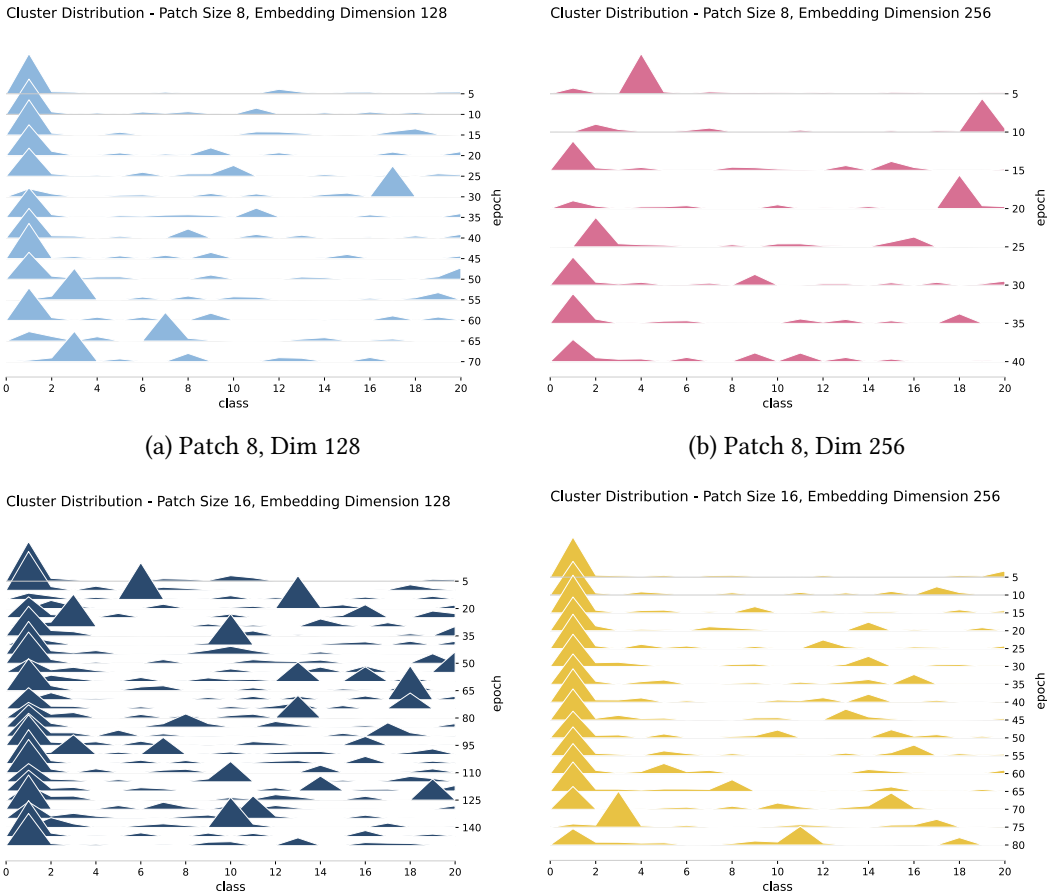
The Bag of Visual Words (BoVW) pipeline uses the pre-trained DINO model to extract patch-level features from molecular junction traces, then encodes each trace as a fixed-size histogram over a learned visual vocabulary. Because storing embeddings for every patch in the full dataset would be prohibitive—the training set of 400 000 traces produces roughly 244 million patches, requiring around 250 GB of RAM at 256 dimensions per patch —a random subset of traces is used to fit the visual vocabulary. Each patch is encoded as a 256-dimensional float32 vector ($256 \times 4 = 1024$ bytes), so even the validation set alone demands around 26 GB. These sampled features are fed into a K-means algorithm to build a "visual vocabulary" (the codebook) of prototype patterns (e.g., 1024 visual words).

Once the codebook is generated, the pipeline processes the entire dataset incrementally. The DINO model extracts features for every patch in a trace, and each feature is quantized by assigning it to the nearest prototype in the visual vocabulary. This process converts variable-length sequential traces into fixed-size frequency histograms of "visual words."

4.4.1 TF vs. TF-IDF Weighting

Before the final global clustering step, the generated histograms are normalised and weighted. The pipeline supports two primary modes for this step: Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF).

Term Frequency (TF) simply uses the raw counts of how often each visual word appears within a single trace, L_2-normalised. While effective for general grouping, TF treats all visual words equally, meaning ubiquitous background noise or common trace features can overshadow infrequent but important structural events.



(a) Patch 8, Dim 128

(b) Patch 8, Dim 256

(c) Patch 16, Dim 128

(d) Patch 16, Dim 256

Figure 4.7: Comparison of cluster distribution evolution (waterfall plots) for different patch sizes (8 and 16) and embedding dimensions (128 and 256).

TF-IDF is specifically used to highlight and cluster rare events. It multiplies the Term Frequency by a global Inverse Document Frequency (IDF) weight. The IDF for a visual word is calculated as $\log\left(\frac{N}{1+\text{DF}}\right)$, where N is the total number of traces and DF (Document Frequency) is the number of traces that contain that specific visual word.

- Common words (high DF) receive an IDF weight close to zero, effectively suppressing background noise.
- Rare words (low DF) receive a high IDF weight, amplifying their signal.

As in TF mode, the resulting TF-IDF histograms are L_2-normalised. This ensures that the global K-means clustering algorithm focuses on the direction in feature space—grouping traces together based on shared rare signatures rather than the overall density of the trace.

4.5 Similarity Search Implementation

A similarity search pipeline was developed to isolate specific molecular events across the entire dataset. The pipeline consists of the following steps:

1. **Feature Extraction and Reference Selection**
   The process begins by loading a pre-trained DINO backbone and the complete valida-

tion dataset (comprising over 41 000 traces). A specific "reference signature" is defined by selecting a target trace and patch index. The DINO model is used to extract a 256-dimensional embedding vector for this reference patch, which serves as the numerical representation of the molecular junction behaviour we aim to isolate across the entire ensemble.

2. **Batched Global Similarity Search**
   Because storing embeddings for all 25 million patches in memory would exceed typical hardware limits (26 GB), the search is performed using a streaming, batched approach. The validation dataset is processed in segments; for each batch, patch embeddings are computed and their cosine similarity to the reference vector is calculated via dot product (as the vectors are L_2-normalised). This allows the system to identify matches without ever holding the full feature matrix in RAM.

3. **Classification**
   Once the similarities are computed, a two-tier filtering system is applied: first, individual patches are flagged if their similarity exceeds a threshold (e.g., $S \geq 0.80$). Second, entire traces are classified as "Matched" if they contain at least N recurring instances of these similar patches. The dataset is then split into two distinct `TraceCollection` objects: the matched ensemble and the remaining traces.

Chapter 5

Results

5.1 Bag-of-Visual-Words

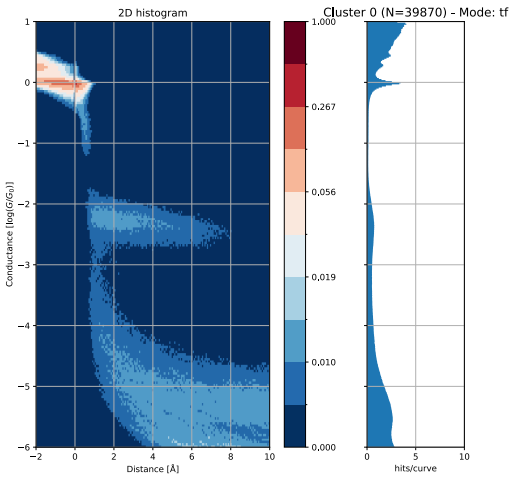
5.1.1 DINO in Bag of Visual Words (BoVW) Clustering

The BoVW pipeline described in Chapter 4 was applied to the full validation set, producing one frequency histogram per trace over a vocabulary of 1024 visual words. The central question is whether these histograms carry sufficient discriminative information to separate traces by physical regime — in particular, to distinguish molecular junction traces from background tunnelling events.

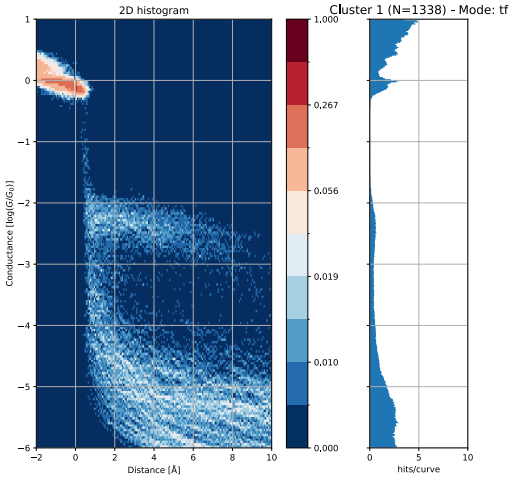
5.1.2 TF vs. TF-IDF Weighting

Given the structure of break-junction traces — dominated by bulk and limit segments — TF-IDF weighting was expected to outperform plain TF: by suppressing these high-frequency, ubiquitous visual words, it should amplify rare molecular-junction patterns and produce better-separated clusters. TF, by contrast, was expected to perform better on the junction segment alone, where all patch types are roughly equally represented.

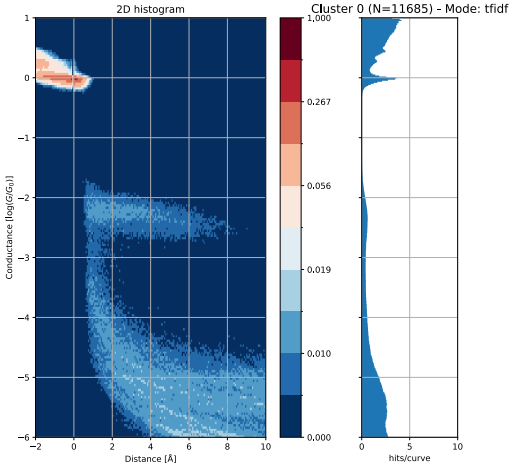
As shown in Figure 5.1, however, both schemes produce near-identical cluster assignments. This convergence is itself a meaningful finding: the DINO embeddings capture the morphology of individual patches — as the similarity search in Section 5.2 will confirm — but aggregating them into a bag-of-words histogram discards the sequential and positional structure that distinguishes a molecular junction trace from a tunnelling-only trace. Since stochastic break-junction events do not repeat at fixed positions, two traces carrying the same physical event may share very few identical visual words; no frequency-weighting scheme can compensate for this.



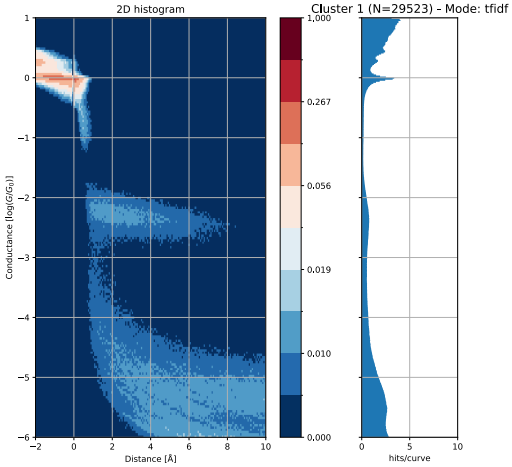
(a) TF Cluster 0



(b) TF Cluster 1



(c) TF-IDF Cluster 0



(d) TF-IDF Cluster 1

Figure 5.1: Comparison of clustering results using TF and TF-IDF weighting schemes. The 2D histograms illustrate the resulting clusters for both weighting methods with $k = 2$ clusters.

5.1.3 Limitations of BoVW Clustering

Within each trace, the K-means assignment in Figure 5.2 broadly reproduces the segmentation an annotator would draw — bulk, plateau, snap-back, and tunnelling regions emerge as coherent contiguous blocks. Two failure modes remain. First, the vocabulary does not maintain class identity **across** traces: the bulk segment is correctly isolated in all three traces shown, yet each trace's bulk patches are assigned a different visual-word class, so there is no single "bulk" word shared across the vocabulary. Second, within a single trace, physically distinct segments are occasionally collapsed into one class — in trace 22208 the after-snapback and reconnection regions are both coloured light green despite occurring at very different conductance levels. The first failure is the decisive one for BoVW: two traces carrying the same physical event produce histograms over different visual words, so no frequency-weighting scheme can recover their similarity.

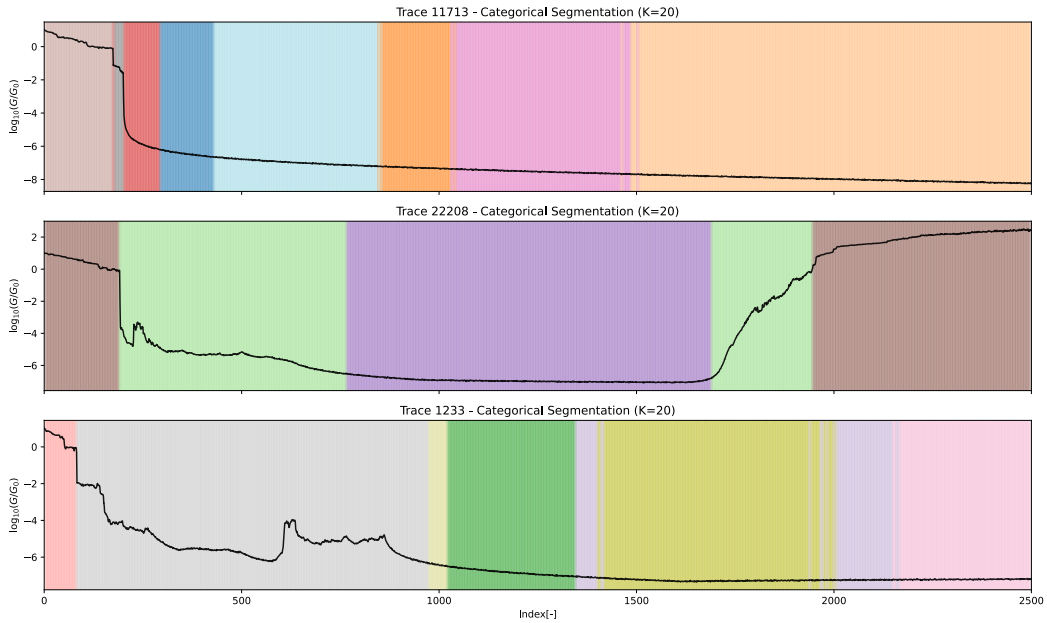


Figure 5.2: K-means clustering of patch embeddings, demonstrating the segmentation of the feature space into distinct visual words.

When K-means assigns patches to clusters based on their embedding vectors, it tracks how far the assigned patches are from their centroid — the mean of the cluster. This distance is quantified by the inertia, or Within-Cluster Sum of Squares (WCSS), defined as

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2, \tag{5.1}$$

where k is the number of clusters, C_i is the set of patches in cluster i, and μ_i is the centroid of cluster i, x is a patch embedding in cluster i.

To optimise the global clustering step — selecting the vocabulary size k for the BoVW encoding — we plot the WCSS for varying k and look for an "elbow" where the rate of decrease sharply slows down (Figure 5.3). This application of the elbow method targets vocabulary size selection specifically, and is separate from the K-means inertia tracked during DINO training (Section 4.3.3.1), which monitored patch-embedding compactness as a proxy for representation quality.

Figure 5.3 shows no discernible elbow: the WCSS decreases nearly linearly across the entire tested range. This absence is consistent with the continuous, stochastic character of molecular conductance transitions. Unlike discrete visual categories in natural images, break-junction events do not form a small set of canonical prototypes — the same molecular junction can produce slightly different plateau shapes across repetitions, and conductance transitions are physically smooth rather than sharply typed. The patch embedding space therefore reflects this continuity: patches are distributed densely and near-uniformly, without the compact, well-separated cluster structure that would produce a clear elbow. Forcing such data into a fixed visual vocabulary discards precisely the variability that carries physical information.

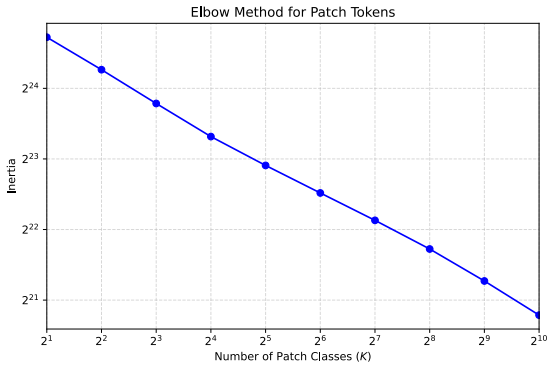


Figure 5.3: Elbow plot for determining the optimal number of visual words for the BoVW clustering algorithm.

Schubert [32] argues that the elbow method can be misleading and that K-means may not find an optimal partition for all datasets. In natural image recognition, Hénaff et al. [33] demonstrate that K-means can discover discrete visual objects — but only in domains where objects have canonical appearances. Break-junction conductance data occupies the opposite regime, where the signal is a continuous physical process rather than a collection of distinct visual categories.

5.2 Similarity search

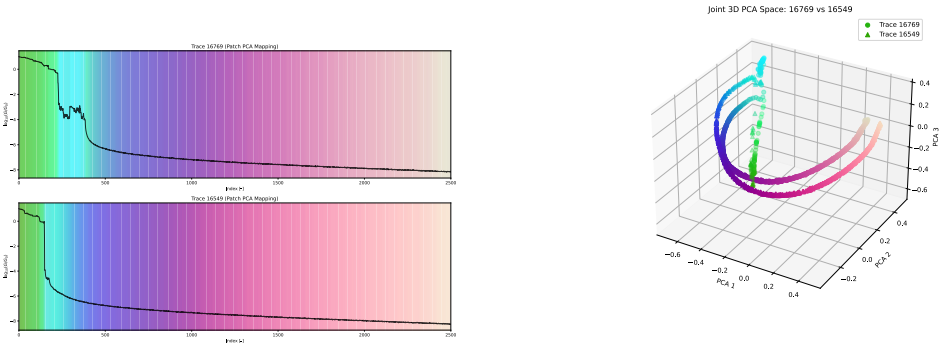
The BoVW result reveals a limitation of histogram-based aggregation, not of the DINO backbone itself. Figure 5.4 confirms this: projecting the full patch embedding space onto its first three principal components — with each component mapped to an RGB colour channel — shows that distinct physical segments of a trace (bulk, molecular plateau, tunnelling current) occupy geometrically separated regions. The embedding space encodes the physics of the measurement even though a bag-of-words summary cannot.

This structure enables a direct, label-free strategy for molecular event isolation. Rather than summarising a trace as a histogram, we select a single reference patch from a known molecular trace — specifically patch 85 of Trace 16769, located within the molecular junction plateau — and scan the dataset for patches with high cosine similarity. Any trace accumulating at least 10 matching patches is classified as a molecular candidate.

To evaluate the feature consistency of our model, we calculate the mutual cosine similarity between patch embeddings. For any two patches A and B, the mutual similarity $S_{A,B}$ is defined as the cosine similarity of their respective embedding vectors $\boldsymbol{e}_A, \boldsymbol{e}_B \in \mathbb{R}^d$:

$$S_{A,B} = \text{cosine}(\boldsymbol{e}_A, \boldsymbol{e}_B) = \frac{\boldsymbol{e}_A \cdot \boldsymbol{e}_B}{\|\boldsymbol{e}_A\|_2 \, \|\boldsymbol{e}_B\|_2 + \varepsilon} \tag{5.2}$$

where ε is a small numerical stabiliser (e.g. 10^{-8}) added to prevent division by zero. Figure 5.5 demonstrates that the model captures distinct segments within traces using this metric. Plotting the cosine similarity of every patch against every other patch within the same trace yields a self-similarity matrix (Figure 5.6). A distinctive "similarity square" on the diagonal — visible just before patch 100 — indicates the presence of a molecular junction.



(a) Trace comparison (b) Joint PCA space

Figure 5.4: PCA visualisation of the patch embedding space, showing the joint 3D distribution and a comparison of specific traces.

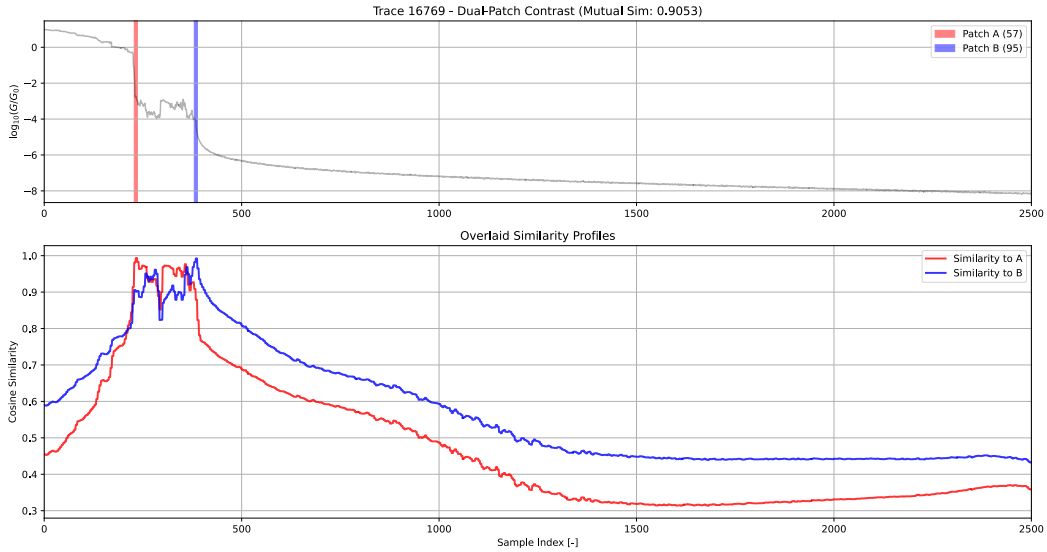


Figure 5.5: Demonstration of internal contrast for Trace 16769, illustrating the mutual cosine similarity between different patches of the same trace.

While a reference signature must be selected manually, the approach is inherently machine-agnostic. Previously used methods relied on selecting a specific conductance range, which is highly instrument-specific due to variations in calibration and hardware sensitivity. The similarity search instead operates on feature morphology, enabling molecular event isolation across different experimental setups.

As illustrated in Figure 5.7, while Figure 5.7b still contains some molecular signatures, the matched traces in Figure 5.7a are notably free of tunnelling-only current. This indicates high precision in the search process, although the recall remains incomplete due to several factors. First, the search relies on proximity to a single anchor patch; however, molecular features occupy a relatively broad region in the feature space (as Figure 5.10 will show), and a single reference may not encompass all variations. Second, to avoid false positives from stochastic similarities in blank traces, we require a minimum of 10 matching patches per trace. This strict threshold likely filters out some valid molecular events, suggesting

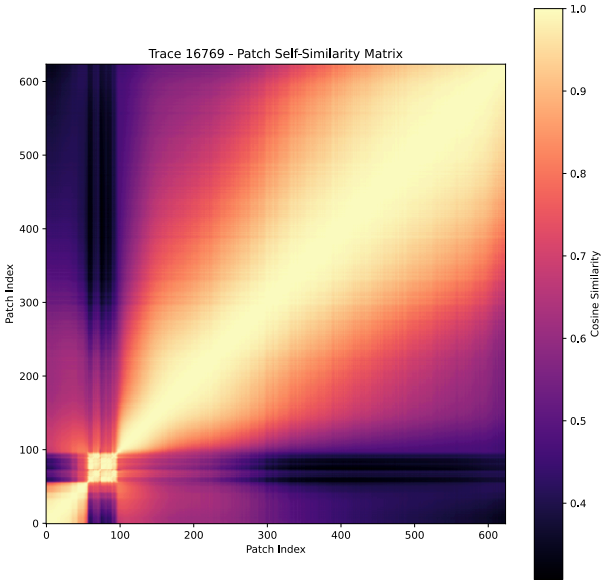
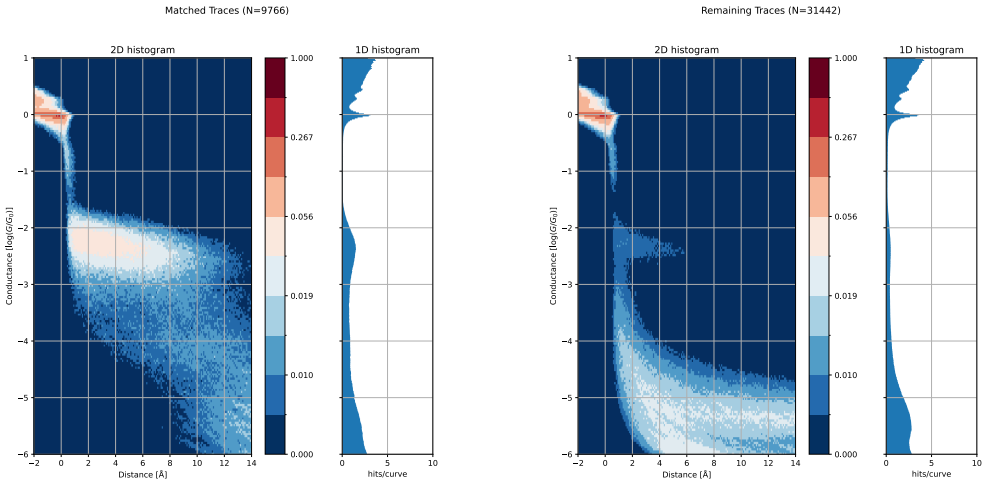


Figure 5.6: Self-similarity matrix for Trace 16769. The matrix highlights the cosine similarity between all patch embeddings within a single trace.

that a more relaxed criterion or the use of multiple anchor patches could further improve the detection rate.

To quantify these results, Table 5.1 reports the number of matched traces across a range of similarity thresholds. At the operating threshold of $S \geq 0.80$, 9 935 out of 41 208 validation traces (24.1%) are classified as molecular candidates. The monotonic decrease from



(a) Matched patches (b) Remaining patches

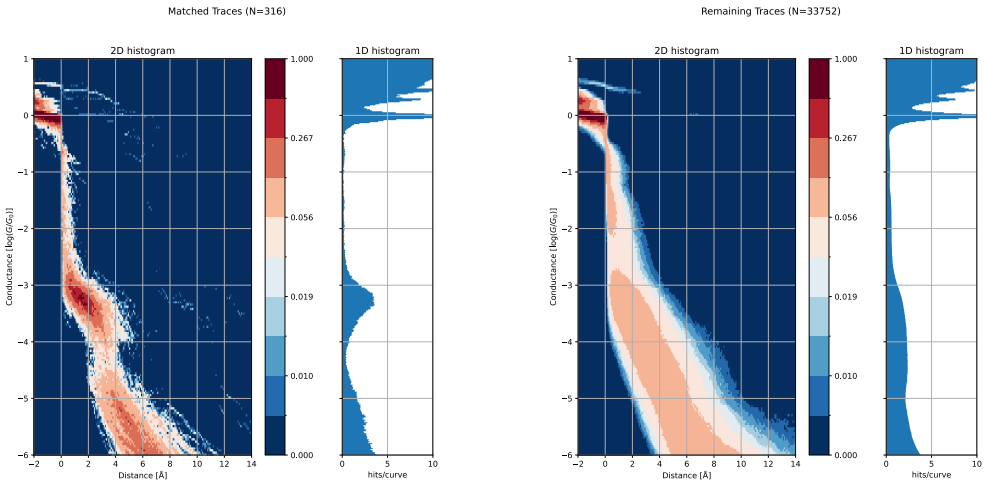
Figure 5.7: Side-by-side comparison of 2D histograms for matched patches and the rest of the validation dataset, illustrating the efficacy of the similarity search.

Threshold S	Matched traces	Percentage
≥ 0.70	18 043	43.8%
≥ 0.75	13 233	32.1%
≥ 0.80	9 935	24.1%
≥ 0.85	7 074	17.2%
≥ 0.90	4 152	10.1%
≥ 0.95	1 529	3.7%

Table 5.1: Threshold sensitivity of the similarity search (Trace 16 769, patch 85, MIN_PATCHES $= 10$, $N = 41\,208$ validation traces).

43.8% at $S = 0.70$ to 3.7% at $S = 0.95$ confirms a meaningful precision–recall trade-off: lower thresholds recover more traces at the cost of including weaker signatures, while higher thresholds restrict the matched set to the closest morphological matches. The full search over 41 208 traces completed in 652.8 s (15.8 ms per trace) on an Apple Silicon device, demonstrating that the pipeline is practical for routine laboratory use without requiring dedicated server hardware.

The generalizability of this approach is further demonstrated on an individual measurement series. Figure 5.8 shows the results for the 1954 dataset, where the same similarity thresholding approach effectively isolates the molecular signature from the background tunnelling traces. The remaining traces might contain other molecular signatures or artefacts that can be further extracted by applying the same approach iteratively.



(a) Matched traces (b) Remaining traces

Figure 5.8: Side-by-side comparison of 2D histograms for matched and remaining traces from the 1954 dataset, demonstrating the specificity of the similarity search on a per-measurement basis.

Segment	Threshold S	Reference patch
Bulk	≥ 0.80	30
Molecular signature	≥ 0.90	85
Tunneling current	≥ 0.95	96
Limit	≥ 0.65	300

Table 5.2: Class-specific similarity thresholds and reference patches used for trace segmentation.

5.3 Segmentation capabilities

As discussed in Section 5.2, and demonstrated in Figure 5.5 and Figure 5.6, the model is capable of identifying and segmenting not only molecular signatures within traces, but also other important segments like bulk or tunnelling current. We illustrate these segmentation results in Figure 5.9. These segments were obtained by comparing patch embeddings against specific reference patches using cosine similarity, with class-specific thresholds as listed in Table 5.2.

The bulk segments are readily identifiable, corresponding to conductance values above zero on a logarithmic scale. Similarly, the limit segments consist of points below a logarithmic conductance of −6, an empirically chosen threshold. These straightforward assignments allow us to verify that the model accurately distinguishes between the bulk and limit regimes.

Snap-back events also cluster together in the feature space with only a few outliers, further demonstrating the model's ability to extract consistent representations of specific physical phenomena.

Another compelling observation is the DINO model's capability to identify the tunnelling current. This segment is crucial for determining the displacement of the electrodes. In a

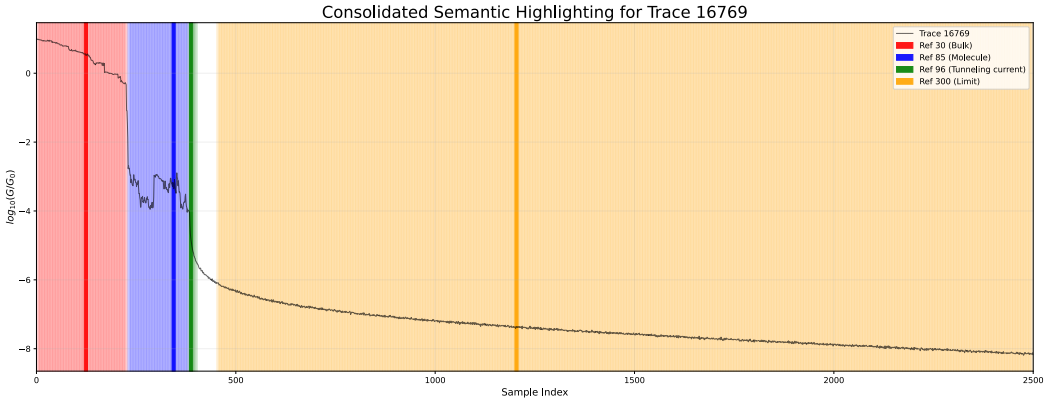


Figure 5.9: Demonstration of the model's segmentation capabilities, showing how different segments of a conductance trace—such as bulk, molecular signatures, and tunnelling—are effectively identified through patch embeddings.

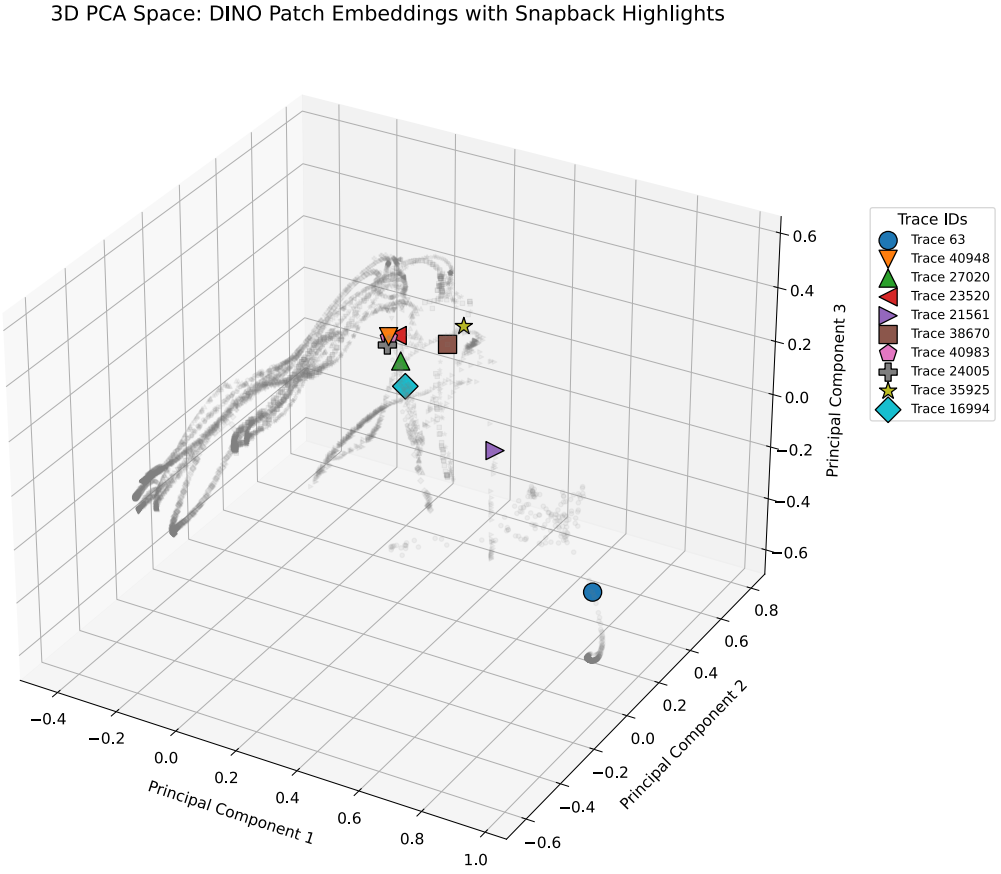


Figure 5.10: 3D PCA visualisation of the patch embedding space, highlighting the snapback region.

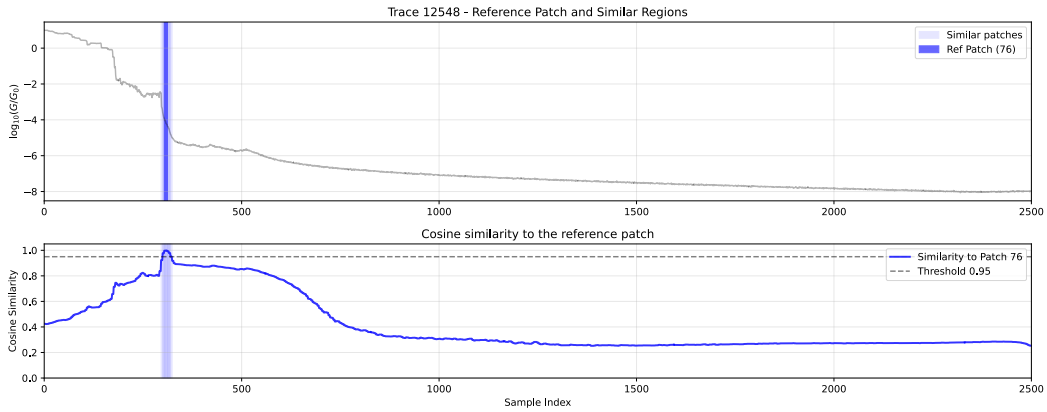
vacuum, the tunnelling current manifests as a linear region with a characteristic slope of one ångström per order of magnitude drop in conductance. Reliable identification of this segment therefore enables the direct calculation of electrode displacement, quantified as the indifactor — a scaling coefficient derived from the tunnelling current slope (see Appendix C). While our recent work has focused on this application, it previously relied on a supervised U-Net segmentation model [34] to isolate the tunnelling current.

As illustrated in Figure 5.11, the similarity search effectively identifies tunnelling current segments within conductance traces. Notably, this robust identification is achieved despite the segment's limited spatial extent, which typically spans only a few overlapping patches (approximately 24 data points).

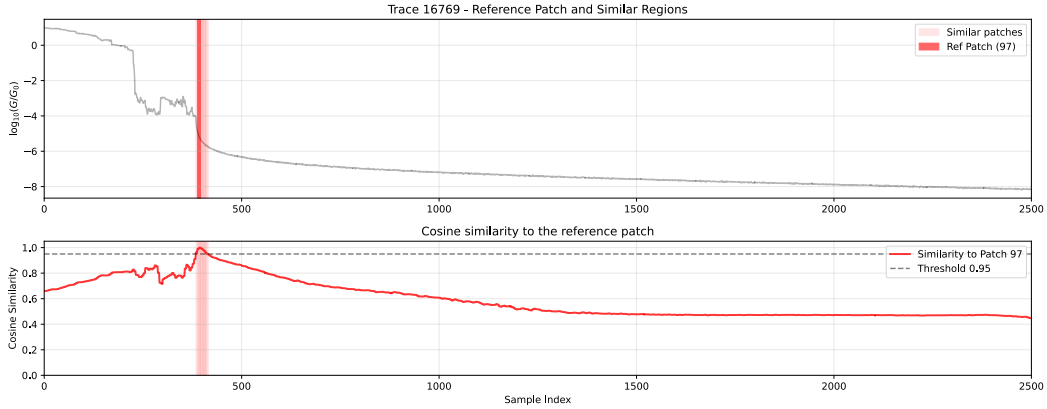
Figure 5.12 confirms that tunnelling patches cluster together in the embedding space, consistent with the similarity profiles above.

5.4 Cross-instrument validation

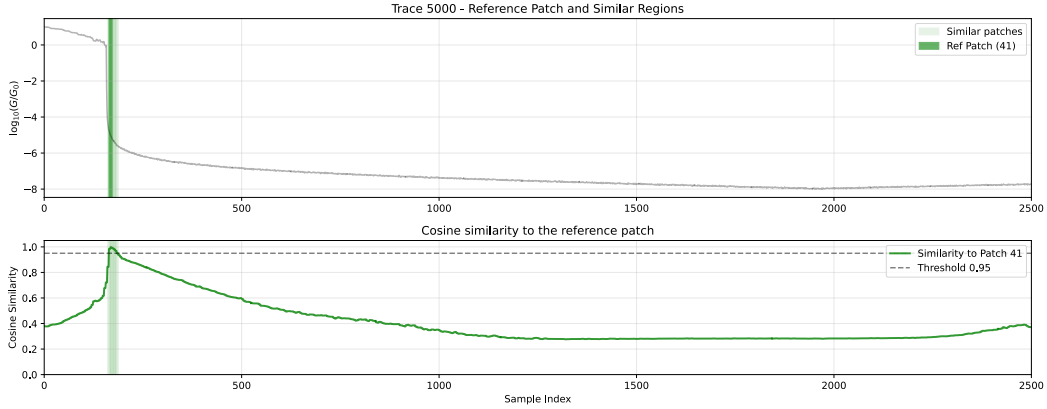
To probe whether the learned embeddings encode rig-specific artefacts or genuine molecule physics, we applied the iCluto-trained DINO model — without any retraining — to the publicly available bp4k single-molecule break-junction dataset from the University



(a) Trace 12548, trace with molecule.



(b) Trace 16769, trace with molecule.



(c) Trace 5000, blank trace with no molecule.

Figure 5.11: Demonstration of DINO's capability to identify tunnelling current segments in conductance traces by comparing the cosine similarity of its embeddings with those of reference patches for tunnelling current.

of Copenhagen [35]. The dataset comprises 5 475 traces of 4 229 data points each, with ground-truth labels of 1 863 Molecule, 3 219 Background, and 393 Noise traces (the noise class is excluded from the metrics below). The only adaptation required was a per-

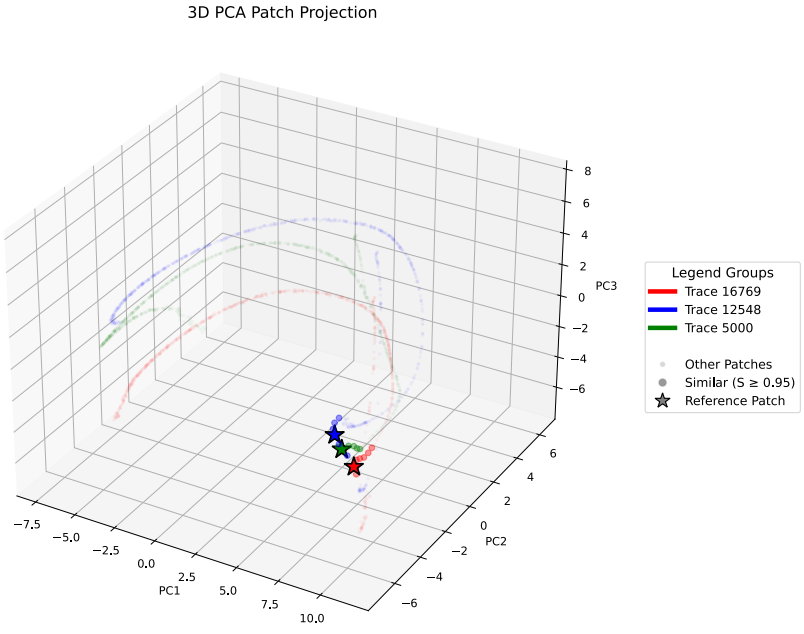


Figure 5.12: Red, green and blue points represent the same traces as in Figure 5.11. Stars highlight the reference patches; patches with cosine similarity above 0.95 to the tunnelling reference are highlighted with larger points.

instrument resampling step: each trace was unified after the conductance limit, aligned at bulk-contact onset, and the molecular bridge rescaled to 180 DINO patches (724 samples) to match the model's effective resolution, then padded to $\mathrm{TRACE\_LEN} = 2\,500$.

Classification reused the similarity-search rule from Section 5.2: a single reference patch (trace 2, patch 145, label Molecule) defines the prototype, and a trace is labelled "Molecule" if at least 10 patches exceed the cosine-similarity threshold.

With ground-truth labels available, every predicted trace falls into one of four bins: true positive (TP, predicted molecule and truly molecule), false positive (FP, predicted molecule but truly background), false negative (FN, missed molecule), and true negative (TN, correctly identified background). Performance is then summarised by

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \text{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{5.3}$$

Precision answers "of the traces I called molecule, how many really are?" — it penalises false positives. Recall answers "of the molecule traces present, how many did I find?" — it penalises false negatives. The F1 score is their harmonic mean, which is high only when both are high, making it the natural single-number summary for a precision–recall trade-off controlled by a threshold. Accuracy ($(\mathrm{TP}+\mathrm{TN})$ over all predictions) is reported alongside it for context. Table 5.3 reports all four metrics across thresholds.

Three operating points are worth noting. The balanced regime at threshold 0.80 achieves $\mathrm{F1} = 0.78$, precision 0.70, recall 0.87. A high-precision setting at threshold 0.95 yields precision 0.99 at recall 0.46 — practically zero false positives, suitable for curating clean

Thr.	#Pred	Precision	Recall	F1	Accuracy
0.70	3 009	0.59	0.96	0.73	0.74
0.75	2 617	0.66	0.92	0.77	0.79
0.80	2 310	0.70	0.87	0.78	0.82
0.85	2 051	0.74	0.81	0.77	0.82
0.90	1 642	0.78	0.69	0.73	0.81
0.95	877	0.99	0.46	0.63	0.80

Table 5.3: Threshold sensitivity on the bp4k cross-instrument dataset (reference patch: trace 2, patch 145; MIN_PATCHES = 10; $N = 5\,082$ labelled Molecule/Background traces).

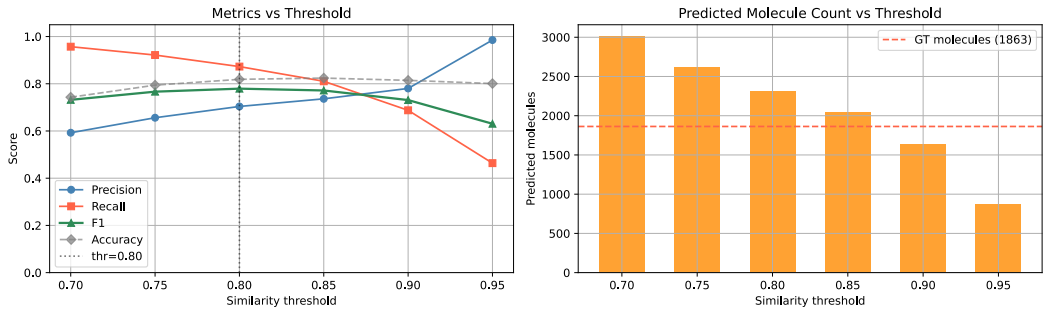


Figure 5.13: Cross-instrument validation on the bp4k dataset (Univ. Copenhagen). The iCluto-trained DINO backbone, applied without finetuning, separates molecule from background traces with F1 ≈ 0.78 at threshold 0.80.

molecule examples. A high-recall setting at threshold 0.70 catches 96% of molecule traces and is well-suited as a first-pass filter.

The key finding is that a single patch embedding from one molecule trace acts as a transferable prototype across instruments. With no finetuning — only a per-instrument resampling step — the same backbone separates molecule from background with F1 $\approx$ 0.78 and accuracy ≈ 0.82 on data the model never saw during training. This indicates that the DINO representation captures instrument-invariant features of the molecular signature rather than rig-specific artefacts of the iCluto setup, supporting the broader claim that the embedding space encodes the underlying physics of the measurement.

Together, these results establish that the quality of the DINO patch embeddings is not the limiting factor in the BoVW pipeline — histogram aggregation is. The PCA projections, self-similarity matrices, and cosine-similarity profiles all confirm that the embedding space encodes the physical structure of break-junction traces in a geometrically meaningful way. Exploiting this structure directly, without a histogram intermediary, yields a label-free molecular event detector that isolates 9 935 candidates from 41 208 validation traces and a segment identifier that matches or replaces prior supervised approaches

across all major trace regimes. The broader implications of these findings and directions for future work are discussed in Chapter 6.

Chapter 6

Conclusion

Mechanically Controllable Break Junction (MCBJ) experiments generate large datasets of stochastic conductance traces in which molecular junction events are rare, transient, and easily obscured by tunnelling background and bulk signals. Traditional histogram-based analysis and linear dimensionality reduction methods such as PCA aggregate trace-level information and struggle to preserve these low-probability signatures. Supervised deep learning alternatives require costly expert-annotated training sets that are impractical to produce at the scale of modern MCBJ datasets.

This thesis proposed adapting DINO — a self-supervised vision transformer trained via self-distillation — to one-dimensional break-junction traces. By treating fixed-length conductance patches as the basic unit of analysis, a 1D DINO backbone learns patch embeddings directly from unlabelled data, requiring no conductance-range hyperparameters and no annotated examples.

Two downstream applications were evaluated. A Bag-of-Visual-Words (BoVW) pipeline encoded each trace as a frequency histogram over a learned vocabulary of patch prototypes, but failed to separate molecular from tunnelling-only traces: the elbow method revealed no natural vocabulary size, TF and TF-IDF weighting produced near-identical cluster assignments, and no frequency-weighting scheme can recover the sequential and positional structure that distinguishes trace classes once it has been discarded by histogram aggregation. A batched cosine-similarity search in the embedding space proved far more effective: distinct physical regimes — bulk, molecular plateau, and tunnelling current — occupy geometrically separated regions, enabling label-free isolation of molecular candidates by proximity to a single reference patch. The approach is machine-agnostic, operating on feature morphology rather than absolute conductance thresholds, and extends naturally to segmentation tasks including tunnelling-current identification, offering a promising annotation-free alternative to the supervised U-Net model used in our prior work.

6.1 Revisiting the Research Goals

Our prior work [8] demonstrated that PCA-based dimensionality reduction, while effective when conductance-range hyperparameters were carefully tuned, systematically lost rare molecular events by collapsing stochastic signatures into an unresolvable region of the latent space. Three limitations motivated the present work: the dependence on expert-chosen conductance thresholds, the inability of linear projections to preserve low-probability features, and the reliance on supervised labels for any model that went beyond clustering.

All three are addressed by the proposed framework. The patch-based DINO backbone operates on feature morphology rather than absolute conductance values, eliminating the need for instrument-specific threshold calibration: the same trained model correctly segments bulk, molecular plateau, and tunnelling-current regions across independent

datasets. The non-linear Transformer encoder preserves the geometric separation of physical regimes in the embedding space — confirmed by PCA projections showing bulk, molecular, and tunnelling patches in clearly distinct regions — where linear PCA on raw traces collapsed these same classes. Finally, the entire representation is learned without any annotation: no labelled traces are required for training, and the downstream similarity search requires only a single reference patch selected interactively by the user.

At an operating similarity threshold of $S \geq 0.80$, the search retrieves 9 935 molecular candidates from 41 208 validation traces in 652.8 s (15.8 ms per trace) on a standard laptop, making it practical for routine laboratory use. The same approach also demonstrates qualitative agreement with the supervised U-Net for tunnelling-current identification, suggesting it as a promising annotation-free alternative, though a quantitative comparison against a labelled reference set remains necessary before any definitive claim of equivalence.

The instrument-agnostic claim is strengthened by an external check on the bp4k dataset from the University of Copenhagen [35]: applied without retraining and using only a per-instrument resampling step, the same iCluto-trained backbone separates molecule from background traces with F1 ≈ 0.78 and accuracy ≈ 0.82 at threshold 0.80, and reaches precision 0.99 at threshold 0.95 (see Section 5.4). That the model transfers between rigs without any finetuning indicates the learned representation captures features of the molecular signature itself rather than artefacts of the iCluto setup.

One limitation remains in the retrieval step: using a single reference patch constrains recall, because molecular features occupy a broad and variable region of the embedding space and a single anchor does not cover all their morphological variants. Extending the search to a small set of complementary anchor patches — selected to span the diversity of known molecular signatures — is the most direct path to improving sensitivity without sacrificing the label-free character of the pipeline.

6.2 Addressing the Failed BoVW Clustering

The BoVW failure is informative rather than merely negative. The PCA visualisation of patch embeddings confirms that the DINO backbone produces a well-structured feature space: distinct physical segments occupy separated regions, and the self-similarity matrix of individual traces reveals the molecular plateau as a coherent diagonal block. The failure lies not in the representation but in the aggregation step. Break-junction events are stochastic — the same molecular junction can produce plateaus of varying length, conductance, and position within a trace. Hard assignment of patches to the nearest vocabulary centroid followed by frequency counting discards exactly the sequential and positional context that differentiates a molecular trace from a tunnelling-only trace. The absence of an elbow in the WCSS plot is a direct consequence of this: patch embeddings are distributed densely and near-continuously, without the compact, well-separated cluster structure that would make a fixed vocabulary meaningful.

Two directions are most promising for overcoming this limitation. The Vector of Locally Aggregated Descriptors (VLAD) [36] replaces hard cluster assignment with the sum of residuals between each patch embedding and its nearest centroid, retaining first-order

distributional information that a frequency histogram discards. Because VLAD encodes how patches deviate from prototype locations — not merely which prototype they are closest to — it is less sensitive to the stochastic variability of break-junction events and has been shown to outperform BoVW in domains with continuous, non-categorical feature distributions. Alternatively, contrastive learning objectives such as SimCLR [28] or supervised contrastive loss applied to pseudo-labels derived from the similarity search could explicitly train the trace-level representation to separate molecular from tunnelling-only classes, bypassing the need for a fixed vocabulary entirely.

6.3 Future Work

6.3.1 Ground Truth Annotation and Quantitative Evaluation

A central limitation of the results presented in this thesis is the absence of ground-truth labels, which prevents a rigorous quantitative evaluation of the similarity search and segmentation pipeline. All assessments currently rely on qualitative inspection of 2D histograms and PCA visualizations. Obtaining precision and recall figures — or even a silhouette score on well-separated clusters — requires a labelled reference set.

The computer vision community benefits from large, standardised benchmarks such as ImageNet [37], which enable direct, reproducible comparison across methods. No equivalent exists for BJ traces: there is no universally adopted, segment-level annotated dataset, and publicly available resources such as Bro et al. [35] provide only trace-level classification labels rather than within-trace segment boundaries. The immediate next step is therefore to produce an in-house annotated dataset — a target of 10 000 or more traces carrying segment-level boundary labels (bulk, molecular plateau, tunnelling current, snap-back) from the validation set would provide a statistically meaningful benchmark for precision, recall, and segment boundary accuracy, and would fill a gap that currently forces all BJ machine-learning studies to rely on indirect or qualitative evaluation.

6.3.2 Adaptive Patch Sizing

The current implementation uses a fixed patch size throughout the trace. Break-junction traces exhibit a pronounced disparity in segment duration: bulk and limit regions span hundreds of data points, while molecular plateaus, snap-back events, and tunnelling-current segments are transient, typically covering only a few tens of data points. A fixed patch size is therefore a compromise — fine enough to resolve these short events, it produces redundant patches in the bulk; coarse enough to cover the bulk efficiently, it may straddle segment boundaries in transient regions. Adaptive patch sizing, where the patch length is chosen locally based on signal variance or a learned boundary detector, could better capture events of varying duration while reducing the number of uninformative patches that the model must attend to.

6.3.3 iCluto as a Research Platform

iCluto has evolved from a clustering application into a general-purpose break-junction analysis platform, integrating trace management, dimensionality reduction, self-supervised representation learning, and interactive annotation under a single command-line interface. The DINO pipeline introduced in this thesis is implemented as a first-class iCluto module, making the similarity search and segmentation capabilities available to

any researcher working with break-junction data without requiring deep learning expertise.

A natural next direction is to treat the trained DINO backbone as a foundational model for break-junction analysis — a fixed feature extractor that downstream tasks (classification, clustering, anomaly detection) can build on without retraining from scratch. Self-supervised architectures related to DINO, such as the Joint Embedding Predictive Architecture (JEPA) [38], offer an alternative training objective that predicts representations in latent space rather than pixel space, which may be particularly well-suited to the smooth, physics-driven structure of conductance traces. Exploring these objectives within iCluto would broaden the range of available pre-trained representations and facilitate systematic comparison on a shared annotated benchmark once one becomes available.

More broadly, this work demonstrates that self-supervised representation learning — developed for natural images — can be adapted to one-dimensional physical measurement data with minimal domain-specific modifications. The resulting pipeline requires no conductance-range calibration, no annotated examples, and no dedicated server hardware, suggesting that the approach could serve as a template for annotation-free analysis in other experimental domains where stochastic, unlabelled time-series data are abundant and expert annotation is prohibitively costly.

6.4 Declaration of AI Usage

During the research and development process of this thesis, artificial intelligence models were utilised as assistive tools. Specifically, the following models were employed: Google Gemini 3 Flash and Gemini 3.1 Pro, and Anthropic Claude Sonnet 4.6 and Opus 4.7 (via Claude Code).

These models were employed for:

- **Software Engineering**: Code refactoring, debugging, and generating optimisation suggestions for the `icluto` library.
- **Utility Code Generation**: Writing boilerplate code, plotting scripts, and SLURM batch scripts for execution on compute clusters.
- **Research Support**: Assisting with literature search, drafting structural outlines, and refining the academic language and tone of the thesis text.

All AI-generated outputs (both code and text) were rigorously reviewed, validated, and modified where necessary. The core scientific ideas, experimental setups, and interpretations of the results remain entirely our own. We take full responsibility for the accuracy, originality, and integrity of the final content presented in this thesis.

Bibliography

[1] A. Aviram and M. A. Ratner, "Molecular rectifiers," *Chemical physics letters*, vol. 29, no. 2, pp. 277–283, 1974.

[2] P. Makk *et al.*, "Correlation analysis of atomic and single-molecule junction conductance," *ACS nano*, vol. 6, no. 4, pp. 3411–3423, 2012.

[3] Z. Balogh, P. Makk, and A. Halbritter, "Alternative types of molecule-decorated atomic chains in Au–CO–Au single-molecule junctions," *Beilstein journal of nanotechnology*, vol. 6, no. 1, pp. 1369–1376, 2015.

[4] J. M. Hamill *et al.*, "Improving single-molecule conductance measurements with change point detection from the econometrics toolbox." [Online]. Available: https://arxiv.org/abs/2401.12769

[5] Z. Balogh, G. Mezei, N. Tenk, A. Magyarkuti, and A. Halbritter, "Configuration-specific insight into single-molecule conductance and noise data revealed by the principal component projection method," *The Journal of Physical Chemistry Letters*, vol. 14, no. 22, pp. 5109–5118, 2023.

[6] F. van Veen, L. Ornago, H. S. van der Zant, and M. El Abbassi, "A generalized neural network approach for separation of molecular breaking traces," *Journal of Materials Chemistry C*, vol. 11, no. 44, pp. 15564–15570, 2023.

[7] F. Huang *et al.*, "Automatic classification of single-molecule charge transport data with an unsupervised machine-learning algorithm," *Physical Chemistry Chemical Physics*, vol. 22, no. 3, pp. 1674–1681, 2020.

[8] O. Klimt, "Break junction data clustering using supervised and unsupervised machine learning," Bachelor's Thesis, Prague, Czech Republic, 2024. [Online]. Available: http://hdl.handle.net/10467/115225

[9] V. Balzani, A. Credi, and M. Venturi, "The bottom-up approach to molecular-level devices and machines," *Chemistry–A European Journal*, vol. 8, no. 24, pp. 5524–5532, 2002.

[10] H. Zhang, J. Li, C. Yang, and X. Guo, "Single-Molecule functional chips: unveiling the full potential of molecular electronics and optoelectronics," *Accounts of Materials Research*, vol. 5, no. 8, pp. 971–986, 2024.

[11] L. Sun, Y. A. Diaz-Fernandez, T. A. Gschneidtner, F. Westerlund, S. Lara-Avila, and K. Moth-Poulsen, "Single-molecule electronics: from chemical design to functional devices," *Chemical Society Reviews*, vol. 43, no. 21, pp. 7378–7411, 2014.

[12] N. Agraıt, A. L. Yeyati, and J. M. Van Ruitenbeek, "Quantum properties of atomic-sized conductors," *Physics Reports*, vol. 377, no. 2–3, pp. 81–279, 2003.

[13] B. Xu and N. J. Tao, "Measurement of single-molecule resistance by repeated formation of molecular junctions," *science*, vol. 301, no. 5637, pp. 1221–1223, 2003.

[14] E. York and L. Venkataraman, "Scanning Tunneling Microscope-Based Break-Junction Technique - A Tutorial," *ACS Physical Chemistry Au*, 2026.

[15] L. Venkataraman, J. E. Klare, I. W. Tam, C. Nuckolls, M. S. Hybertsen, and M. L. Steigerwald, "Single-molecule circuits with well-defined molecular conductance," *Nano letters*, vol. 6, no. 3, pp. 458–462, 2006.

[16] J. Zhao, M. Feng, D. B. Dougherty, H. Sun, and H. Petek, "Molecular electronic level alignment at weakly coupled organic film/metal interfaces," *ACS nano*, vol. 8, no. 10, pp. 10988–10997, 2014.

[17] J. Nejedlý, "The synthesis of *pi*-electron systems suitable for transfer and retention of charges", 2021.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–2.

[20] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[21] A. Kirillov *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International*

Conference on Computer Vision (ICCV), 2023, pp. 4015–4026.

[22] M. Caron *et al.*, "Emerging Properties in Self-Supervised Vision Transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[23] M. Oquab *et al.*, "DINOv2: Learning Robust Visual Features without Supervision." 2023.

[24] O. Siméoni *et al.*, "DINOv3." [Online]. Available: https://arxiv.org/abs/2508.10104

[25] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

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

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16000–16009.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, 2020, pp. 1597–1607.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.

[30] P. Kage, J. Rothenberger, P. Andreadis, and D. Diochnos, "A review of pseudo-labeling for computer vision," *Journal of Artificial Intelligence Research*, vol. 85, 2026.

[31] P. Goyal *et al.*, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," *arXiv preprint arXiv:1706.02677*, 2017.

[32] E. Schubert, "Stop using the elbow criterion for k-means and how to choose the number of clusters instead," *SIGKDD Explor. Newsl.*, vol. 25, no. 1, pp. 36–42, July 2023, doi: 10.1145/3606274.3606278.

[33] O. J. Hénaff *et al.*, "Object discovery and representation networks," in *European conference on computer vision*, 2022, pp. 123–143.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[35] W. Bro-Jørgensen, J. M. Hamill, R. Bro, and G. C. Solomon, "Trusting our machines: validating machine learning models for single-molecule transport experiments," *Chemical Society Reviews*, vol. 51, no. 16, pp. 6875–6892, 2022.

[36] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311. doi: 10.1109/CVPR.2010.5540039.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[38] M. Assran *et al.*, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 15619–15629.

Appendix A

List of abbreviations

BoVW	Bag of Visual Words
BJ	Break-junction
CLS	Class token — special aggregator token prepended to patch sequences in Transformer encoders
CNN	Convolutional Neural Network
CTU	Czech Technical University
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DF	Document Frequency
DINO	Self-Distillation with No Labels
EMA	Exponential Moving Average
GELU	Gaussian Error Linear Unit
HPCC	High-Performance Computing Cluster
HPCg	High Performance Computing group (at IOCB Prague)
IDF	Inverse Document Frequency
IOCB	Institute of Organic Chemistry and Biochemistry
IVC	Current-to-Voltage Converter
JEPA	Joint Embedding Predictive Architecture
MAE	Masked Autoencoder
MCBJ	Mechanically Controllable Break Junction
MesH	Mesitylene (1,3,5-trimethylbenzene), a solvent
MHSA	Multi-Head Self-Attention
MLP	Multi-Layer Perceptron
NMS	Non-Maximum Suppression
PCA	Principal Component Analysis
Pre-LN	Pre-Layer Normalization
RANSAC	Random Sample Consensus
SAM	Segment Anything Model
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform
SimCLR	Simple Framework for Contrastive Learning of Visual Representations
SLURM	Simple Linux Utility for Resource Management
SSL	Self-Supervised Learning
STM-BJ	Scanning Tunneling Microscope Break Junction
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
t-SNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
U-Net	U-shaped Convolutional Network (for biomedical image segmentation)
ViT	Vision Transformer
VLAD	Vector of Locally Aggregated Descriptors
WCSS	Within-Cluster Sum of Squares

Appendix B

iCluto toolkit

iCLUTO (*improved CLUstering TOolkit*) is a command-line application for clustering conductance traces from break-junction experiments. Developed at CTU FEE under IOCB supervision since October 2022, it has gone through several iterations reflecting the evolving needs of Ivo Starý's group. The toolkit integrates feature extraction, dimensionality reduction, and clustering (K-Means, DBSCAN, HDBSCAN, BIRCH) into a single configurable workflow, and ships helper commands for data conversion, merging, 2D-histogram plotting, and interactive cluster inspection.

B.1 Submission notebooks

As a proof of concept, six Jupyter notebooks are provided that demonstrate how iCluto's DINO-based feature extraction works in practice. The notebooks are numbered in narrative order and walk through the full pipeline — from training inputs to unsupervised clustering results.

The artefacts are distributed as a self-contained archive available at thesis.icluto.oklimt.com with the following structure:

```
submission/
  icluto-0.1.10-py3-none-any.whl
  dino_model_epoch30.pth
  dino_model_metadata.json
  traces.npy
  notebooks/
    00_model_card.ipynb
    01_interactive_augmentations.ipynb
    02_dino_segmentation_analysis.ipynb
    03_similarity_search.ipynb
    04_bovw_clustering_pipeline.ipynb
    05_external_validation.ipynb
```

`icluto-0.1.10-py3-none-any.whl`	iCluto package (install once, no internet required)
`dino_model_epoch30.pth`	trained DINO weights used throughout the evaluation
`dino_model_metadata.json`	architecture hyperparameters for the model above
`traces.npy`	conductance-trace dataset (NumPy array)
`notebooks/00_model_card.ipynb`	entry point — verifies artefacts and gives a concise model overview
`notebooks/01_interactive_augmentations.ipynb`	the input augmentation pipeline that drives self-supervised training

`notebooks/02_dino_segmentation_analysis.ipynb`	how the trained model semantically segments traces via attention maps
`notebooks/03_similarity_search.ipynb`	retrieving rare signatures across the dataset by nearest-neighbour search
`notebooks/04_bovw_clustering_pipeline.ipynb`	end-to-end unsupervised clustering with the learned DINO features
`notebooks/05_external_validation.ipynb`	cross-instrument generalisation on the University of Copenhagen bp4k dataset — same backbone, no re-training

B.1.1 Setup

Python 3.11 or newer is required. Download the archive from thesis.icluto.oklimt.com and run the following commands once after extracting it:

```
unzip icluto_dino_submission.zip
cd submission
python -m venv .venv && source .venv/bin/activate   #
Windows: .venv\Scripts\activate
pip install icluto-0.1.10-py3-none-any.whl
pip install jupyter ipywidgets
jupyter lab notebooks/
```

The recommended reading order follows the notebook numbering. `00_model_card.ipynb` serves as the entry point and can act as a standalone model card; the remaining notebooks each cover one aspect of the pipeline and can be run independently once the environment is set up.

Appendix C

Why the Indifactor is Important

This work was presented as a work-in-progress extended abstract at the POSTER 2026 student conference[5].

In break-junction experiments, the precise displacement of electrodes is initially unknown and must be inferred from conductance data. This calibration is essential for generating accurate 2D histograms (e.g., Figure 1.1). The "Indifactor" serves as a dynamic scaling factor to convert sample indices into physical distance (Angströms).

Traditionally, this has been calculated using the **Legacy Indifactor Formula**:

$$f = \frac{g_1}{j_0 s} + \frac{1}{2}\frac{g_2}{j_2 s} \tag{C.1}$$

Where:
- $s = 1.28$ is the empirical slope.
- j_0 is the snap-back index.
- j_2 is the midpoint index ($\frac{j_0}{2}$).
- $g_1 \approx 10$ at index 0 and g_2 is conductance at j_2.

To improve upon this baseline, an automated pipeline using 1D U-Net segmentation and RANSAC fitting was developed to isolate the tunnelling current segment and robustly extract the slope (Figure C.1). This approach leverages the physical basis that one decade of decay in tunnelling current in vacuum is roughly equivalent to 1 Å. In the MesH solution used here, however, the conversion factor is empirically found to be approximately 0.531 Å per decade of decay, calibrated against the legacy indifactor as ground truth.

C.1 Comparative Analysis

The U-Net-based approach demonstrates high slope stability over tens of thousands of traces (Figure C.2).

When compared to the legacy formula, the automated U-Net method provides a significantly narrower distribution of indifactors (Figure C.3a). The standard deviation is reduced from $\sigma \approx 0.0118$ (Legacy) to $\sigma \approx 0.0057$ (U-Net), yielding roughly a 2x improvement in precision. Furthermore, the U-Net approach mitigates an increasing error trend observed in the legacy method as values grow larger (Figure C.3b).

Additionally, investigating the indifactor distributions reveals a notable shift between blank gold samples and molecular datasets like $4'$ -Bipyridine (Figure C.4). This highlights the importance of robust indifactor calibration to distinguish subtle molecular behaviours from baseline variations.

[5] https://poster2026.sciencesconf.org

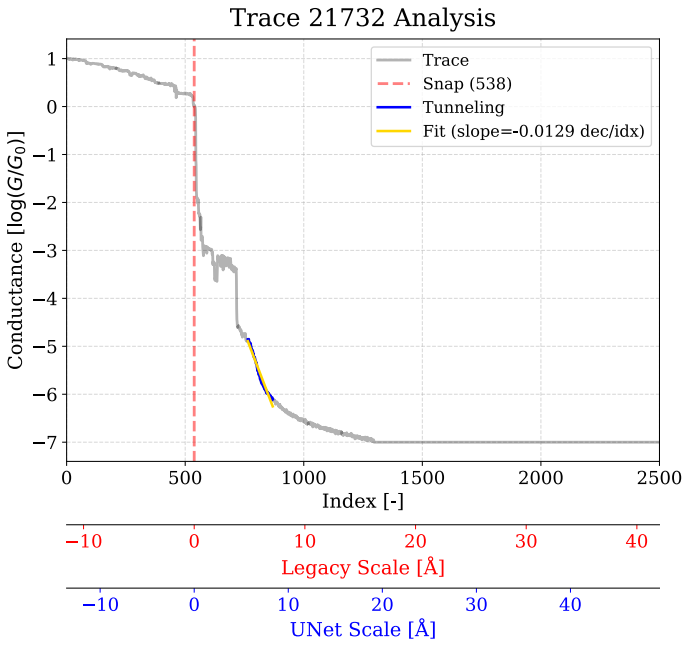


Figure C.1: Trace analysis showing the original trace, the identified snap-back point, and the U-Net-fitted slope.

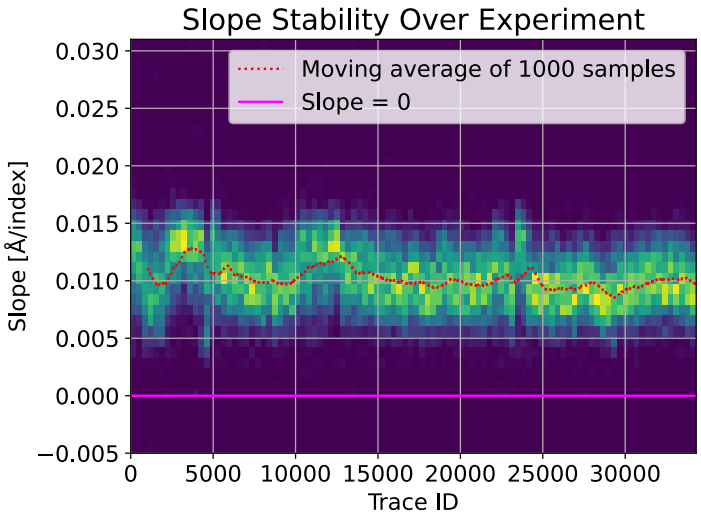
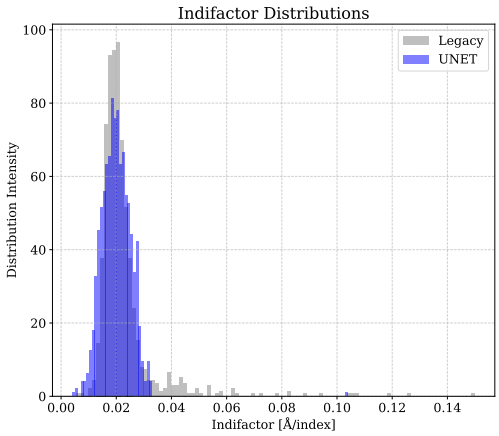
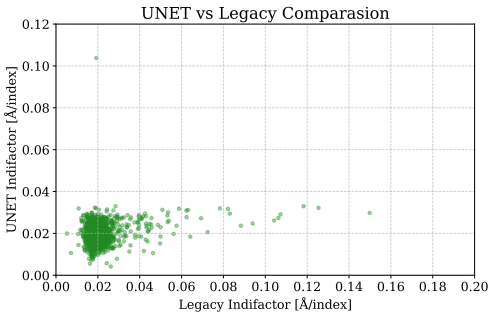


Figure C.2: 2D histogram of U-Net-fitted slopes over 30 thousand traces for 4′ -Bipyridine.



(a) Distributions comparison demonstrating the narrower variance of the U-Net methodology.



(b) Scatter plot demonstrating how the U-Net method mitigates the increasing error trend present in the legacy approach.

Figure C.3: Comparison between the legacy indifactor calculation and the U-Net methodology.

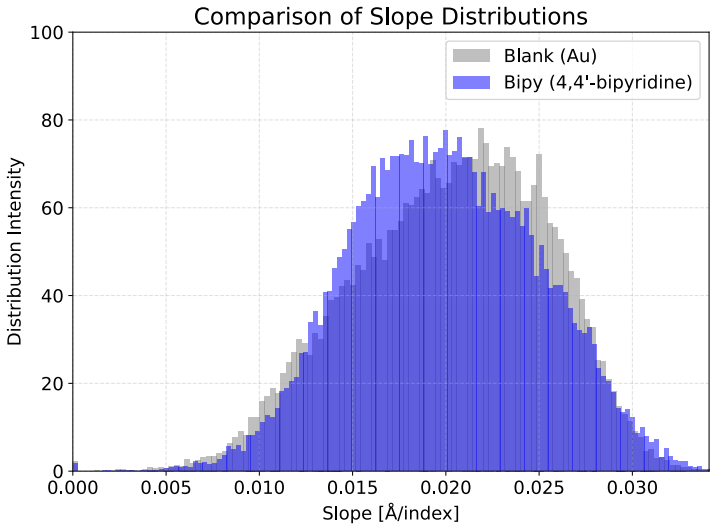


Figure C.4: Distribution of indifactors comparing Blank and 4′ -Bipyridine samples.