CABINET / ARTML

The Ghost in the Gallery

How a point on a painting became a precomputed answer

Gabriel George

Technical white paper
16 September 2026
Revised 17 September 2026

A fixed visual interaction, compiled into an answer table.

Design, experiments, and visual intelligence that finishes its work before the visit.

Contents

Abstract

I wanted to build an art experience that begins with looking: mark a detail in a painting and discover where else something like it appears. The useful systems insight was that this apparently open-ended interaction has a finite query space. A point resolves to a cell on a fixed patch grid. For Cabinet's installed collection of 308 paintings, there are **686,116 possible addresses**. Every answer can be computed before a visitor arrives.

Getting there took three designs. A hosted vision model established the point-and-response interaction. Hand-authored grids explored how much could be prepared in advance. A frozen visual encoder then made it practical to prepare an answer for every point. Each version clarified the next: I wanted the visitor to choose the detail, the answer to invite more looking, and the free product to avoid a model-inference charge on every interaction.

The resulting compiler ranks visual matches offline, keeps up to eight regions from different paintings, and stores their patch IDs. The serving artifact is **14.48 MB**, including geometry and labels. Each lookup reads one **20-byte route record**; no model or vector search runs per query. Independent reference checks verify the compiler's exact ranking within its assigned cluster. A census of all stored answers finds eight destinations for **99.86%** of patches and two for the remaining 0.14%, with no empty answers.

This paper explains the design journey, the compiler, and the measurements that make the result inspectable. New checks use the exact shipped 308-painting corpus to measure seed durability, changes in returned walls, and agreement with low-cost visual baselines. They distinguish a reproducible serving artifact from relationships that may change on recompilation. The reusable contribution is independent of any particular cluster label: **when an interaction has a fixed, enumerable question space, its answers can be shipped without the model that produced them.**

1. The problem

The running example throughout this paper is the **installed release: 308 paintings and 686,116 patches**. Two earlier research settings are identified where they appear: a 310-painting development snapshot and a separate expanded corpus of 1,005 paintings. The new evaluation in Sections 7 and 8 uses the installed release itself. Section 1.4 provides the full evidence map.

1.1 The interaction

I wanted Cabinet to give a person time to look before a title, an artist, or a wall label supplied a way to read the painting. A painting appears with no title, no artist, and no date. The viewer marks the point that interests them. Only then does the system answer, and its answer is not text but more looking: up to eight regions from other paintings that resemble the one marked.

One constraint follows from this, and it drives every decision in the paper:

> The system must answer a question about an arbitrary point in an arbitrary painting, and the answer must be visual rather than linguistic.

"Arbitrary point" is the expensive half. A viewer may mark the centre of a face, a fold of drapery, a stretch of sky, or the blank ground between two figures. The system cannot precompute answers for a handful of interesting places, because deciding which places are interesting is precisely the judgement the product exists to hand back to the viewer.

Two complete stored answers show the interaction. Figure 1 starts at the girl's right eye—the eye on the viewer's left—in Vermeer's *Girl with a Pearl Earring*. Figure 2 retains the earlier Turner sky example. The points were chosen before inspecting their answers, and each plate includes all eight returned regions in rank order. Together they show what a visitor can do with the same small lookup: follow a sharply defined detail or explore a broader visual resemblance.

1.2 The constraints

Constraint	Why it binds
No model-inference charge per query	The product is free; inference billed per interaction would scale with use. Hosting still has a resource cost.
No inference at request time	A hosted model call adds latency, a dependency, a rate limit, and a bill.
Small enough for a hobby-tier host	Vercel's included bandwidth is the budget.
Answers any point, not selected points	See above. This rules out sparse hand annotation.
No text in the matching path	Consulting titles or descriptions would mean retrieving on metadata while appearing to look.
Honest about uncertainty	A resemblance is a measurement, not a claim about influence or meaning.

The third and fourth constraints pull against each other. Resolving that tension is what the final design contributes.

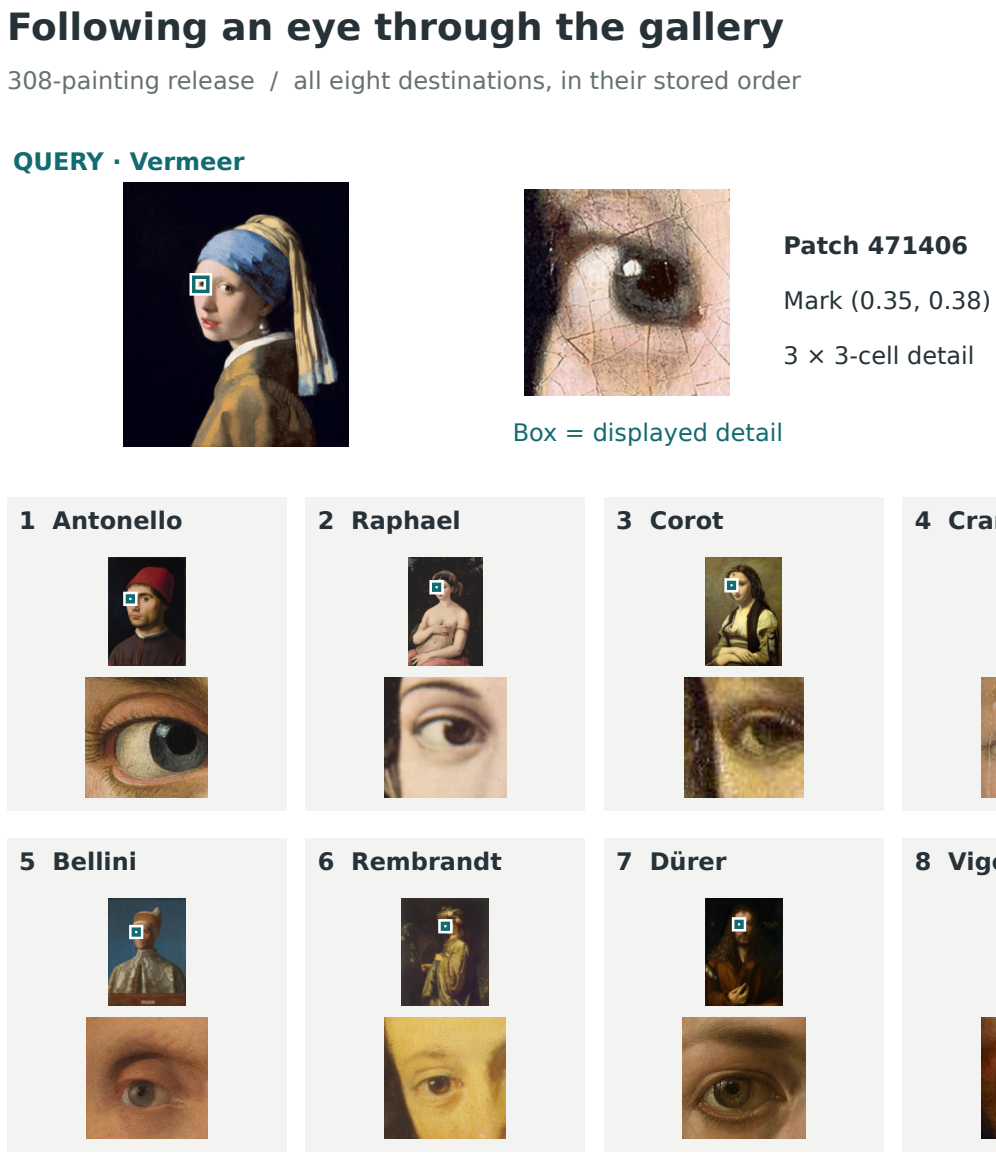


Figure 1: Following an eye through the gallery. The requested point (0.35, 0.38) resolves to patch 471406, row 21 and column 16 on a 48 × 56 grid (zero-based). All eight destinations are the installed release's stored answer, without substitutions. Boxes mark the 3 × 3-cell display neighborhoods; the features also incorporate context beyond those crops. The example makes the interaction visible; it is not a sampled quality benchmark. Corpus: installed release, 308 paintings. Artist names are publication credits added after retrieval; full credits are in Appendix A.

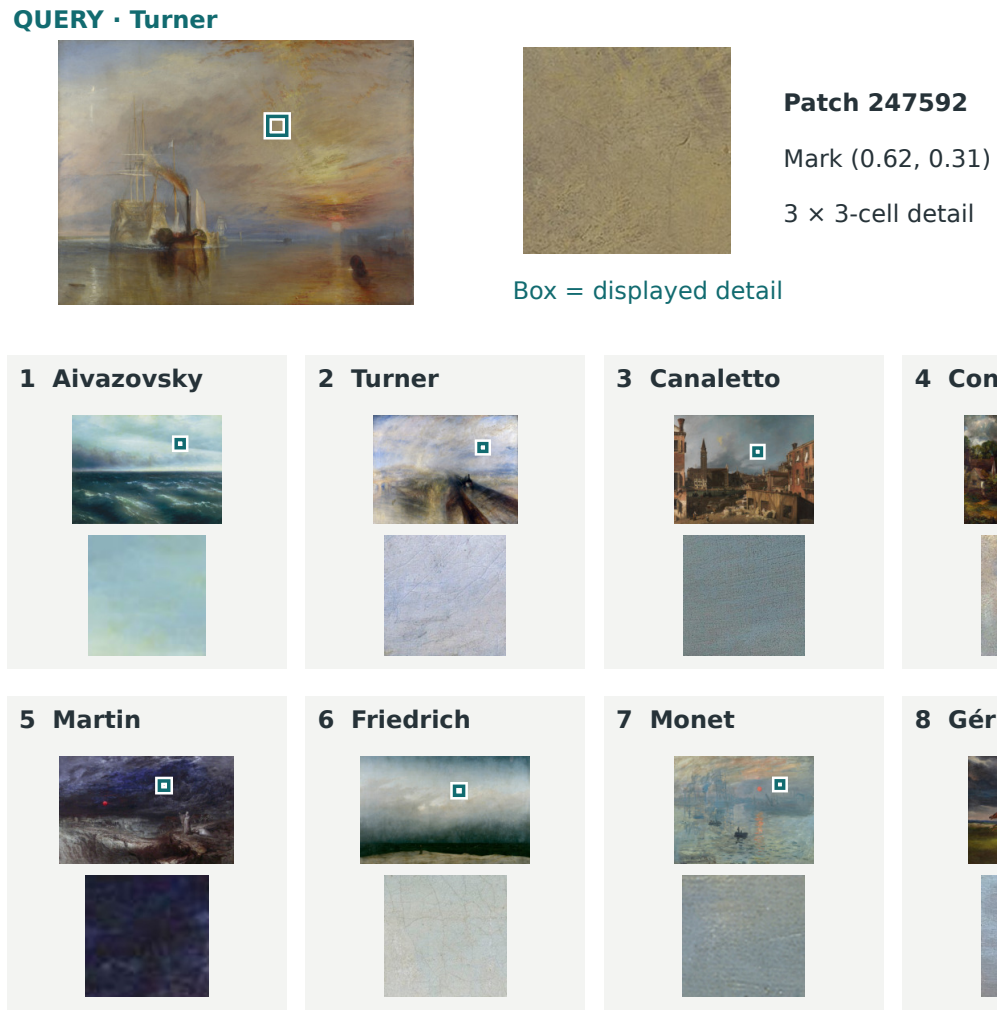


Figure 2: The same interaction in Turner's sky. Point (0.62, 0.31), also used in Figure 4's address diagram, resolves to patch 247592. All eight destinations appear in stored order. Boxes and enlarged crops show the 3 × 3-cell display neighborhoods, while the full paintings supply context. Corpus: installed release, 308 paintings. This is a worked example of retrieval, not a claim about historical influence. Artwork credits and reproduction details are in Appendix A.

1.3 The corpus

The installed collection contains 308 public-domain paintings from Wikimedia Commons, with project records checking each reproduction's reuse terms. Commons serves the images directly. I began with a target of 365, one per day, and removed images with insufficient resolution or unclear rights. Keeping a smaller collection became a deliberate choice: I wanted paintings that could become familiar through repeated visits. That remains an intention to test with visitors.

1.4 Three corpora, and which claim belongs to which

Three corpus sizes appear in this paper. They are close enough in two cases to be misread as the same thing, so they are separated here and every later figure is tagged against this table.

Corpus	Size	What it is	What is measured on it
Installed release	308	One painting per current catalogue case	Compiler agreement and baselines (§7), stored-answer census (§7.2), seed durability and wall overlap (§8)
Historical snapshot	310	An earlier catalogue state, before removals and scan replacements	Sign cross-tab (§3.3), seed durability (§8.1), layer/k sweeps (§9.1, §9.5)
Research	1,005	The expanded research corpus	Backbone comparison and current concept counts (§7.4, §9.2)

1.5 How the project arrived here

I began with a hosted vision model: three marked details produced three responses and a synthesis. It made the interaction tangible, but the recurring cost did not fit a free project. Preparing responses by hand moved that work to authoring time; fine grids then multiplied the effort per painting.

Investigating image embeddings and self-supervised learning led to DINOv3. An AI assistant helped connect the tooling and later supplied descriptive words for reviewed clusters. The step that brought the design together was to keep the relationships the model produced. If all possible marks could be enumerated, the server would only need their answers. The following sections explain how each version contributed to that result.

2. Design one: inference-time intelligence

2.1 What it did

The original implementation, written first in Swift and later ported to JavaScript, invited the viewer to mark up to three points on a painting. It then composed a prompt containing the painting's title, artist, and year; two full-frame images, one with the marks drawn as numbered coloured rings and one clean; the marks' coordinates normalised to the model's 0 to 1000 vision grid; and the painting's hand-authored interest points with their own coordinates and curator notes.

The model returned structured JSON. For each mark it gave `what_it_is` and `meaning`, plus whether the mark landed on an authored interest point, and then one unified paragraph tying the three choices together.

The prompt survives in the repository at `src/lib/ai/prompts.js`, behind a disabled feature flag.

2.2 What was right about it

Three decisions from the prototype became foundations of the current design.

The interaction itself. Mark a point, receive an answer about that exact point. Unchanged across all three designs.

The three-mark affordance. The production API still accepts one to three points per request. The shape outlived the model that motivated it.

The refusal of free text. There was never an "ask the painting anything" box. The only user-originating inputs to inference were image pixels and normalised coordinates. This began as a prompt-injection defence and had the useful side effect of keeping the interaction about looking rather than about conversation.

2.3 What led to the next design

Economics, first. Every interaction invoked a hosted multimodal model with two full-resolution images attached. Per call this is inexpensive. Per product it creates a recurring cost for a product intended to remain free. A payment system was built to address this, comprising anonymous authentication, a credit ledger, checkout integration, five free readings and then 100 for US$4.99. Building it clarified the product decision: the machinery for charging had become substantially larger than the feature. I wanted to spend that complexity on the experience, so I looked for a design whose cost did not track every mark.

Data handling, second. The project also treated provider data-use terms as a constraint on a free inference tier. This was part of the design rationale, not a claim that a hosted model is technically necessary for this interaction.

The answer format mattered too. The model returned prose about meaning. For this product, I increasingly wanted a response that kept the visitor looking. The stated purpose is to give the eye room to form a question before the record supplies an answer. A paragraph of confident interpretation, delivered instantly, is the record arriving early.

The whole stack remains in the codebase, disabled and labelled dormant. It is preserved rather than deleted because re-enabling it is a business decision, not an engineering one.

3. Design two: authoring-time intelligence

3.1 What it did

If the cost sits at inference time, remove inference. Move the knowledge to authoring time and store it.

A painting was divided into a labelled grid, columns A through H and rows 1 through 8, then exported as overview and sub-region tiles. Each region was described by hand. The artifact survives as `grid-authoring/`: 49 directories, one per painting, holding 678 PNG artifacts, including 676 numbered tiles named by grid span (`01-A1-C4.png`, `05-A4-C7.png`) alongside each source image.

Density tracked compositional complexity. Duccio's *Maestà* needed 50 tiles; Holbein's *Ambassadors* and Bosch's *Garden of Earthly Delights* received dedicated treatment; a simpler work such as *Vanitas* took 24.

A longer-lived strand of the same idea also exists. 941 authored hotspots sit across the case files, each a hand-placed coordinate carrying a symbol from a vocabulary of roughly 40 authored

signs, among them *The Gaze*, *The Blade*, and *The Void*, with written lore and cross-painting pattern notes. These remain in the codebase and are still rendered. Section 3.3 explains why.

3.2 Where hand authoring reached its limit

Cost per painting did not fall with scale. Where the model-based design paid an API call per use, this one paid human hours per painting, and human hours do not get cheaper. 49 paintings were completed. The remaining 259 represented the same effort again, five times over.

Fine details need fine resolution. A coarse grid is quick to author, but it cannot reliably distinguish the details a visitor might mark. Told that region C4 holds "figures and drapery," the system cannot distinguish a mark on a hand from a mark on a sleeve. Useful resolution demands a fine grid, and a fine grid multiplies the authoring burden by the square of the refinement.

The obvious fix recursed into a harder problem. The natural response is a variable grid: fine cells over faces and hands, coarse cells over sky and ground. But deciding where the fine cells belong is itself the perceptual problem the system was built to solve. Approximating it requires something like a saliency or depth map, which requires a model, which returns to design one while also retaining a large authoring burden. That would add a model pipeline while retaining much of the authoring work. For this project, the more useful next step was to automate visual matching itself.

Coverage was structurally uneven. Of 310 cases at the time, 102 carried no authored hotspots at all. Those gaps limited where visitors could make a discovery. And because the authored points were the same 941 places for everyone, the design tended toward a shared checklist. I wanted the visitor's own coordinates to determine the path.

3.3 What was learned, and kept

The authored vocabulary was not deleted. A historical Cabinet analysis joined 941 authored hotspots to the frozen 310-painting release and its then-current reviewed labels, with none unresolved. These are historical aggregate results, not a join to the current 308-painting space or the sealed research metadata. "Machine label" below means a word assigned after clustering, not a word produced by DINOv3.

One plausible reading is that a sign records an authored interpretation while a cluster label describes recurring visual context. But the table alone does not establish why they disagree. Common labels may dominate by base rate; hotspot placement, contextual features, and imperfect post-hoc labels may also explain the pattern. In particular, 44 of 161 is 27.3%, not a majority. The result supports keeping the vocabularies distinct, without claiming that the model has understood what the marked detail means.

What the machine vocabulary can do is the thing the authored one could not: hold an opinion about all 686,116 patches. Every painting yields a word at every point. A viewer who marks the sky gets `Sky`, and one who marks the frozen pond in the same picture gets `Water`. This recovers the design goal that design two lost. Collections can differ because they record each viewer's chosen coordinates. Its effect on the visitor experience is a question for a user study.

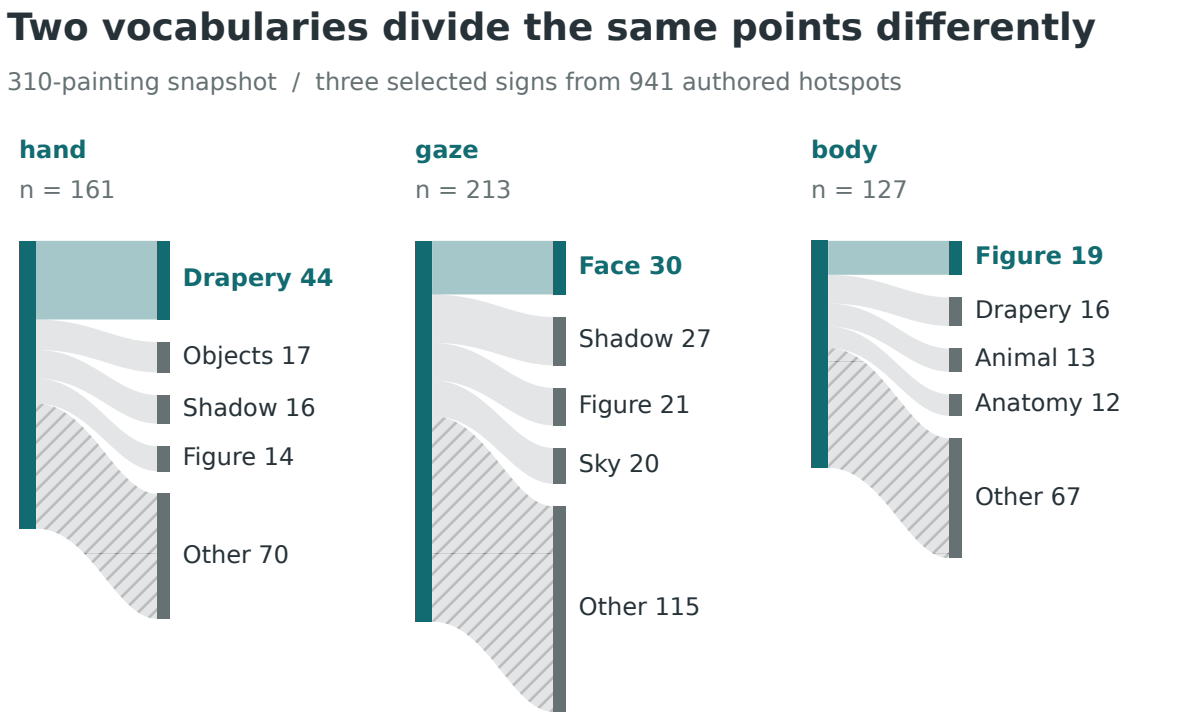


Figure 3: Authored signs and reviewed cluster labels are not interchangeable. Of 161 hand hotspots, 44 map to Drapery; Hand is absent from the published top four, but its exact count is not reported. Ribbon thickness uses the same count scale in all panels, and Other retains the full unreported mass. These are three selected signs from the historical 941-hotspot analysis, not all signs. No patch-frequency baseline or independent annotation control was computed, so the diagram cannot establish a semantic-versus-visual explanation. Source: Cabinet/docs/ECHO-LABELS.md, 310-painting snapshot.

4. Design three: compile-time intelligence

4.1 The reframing

The first two designs made the placement of work explicit: a model call for each interaction, or human authoring for each region. The third moves the visual work to compile time: encode the collection, answer every supported query, and ship the resulting table.

This is viable only when the set of possible questions is finite and known in advance. Here it is, and that observation is what makes the rest of the paper possible:

> A viewer can mark any point in any painting, but a point resolves to a patch, and there are only 686,116 patches in the corpus.

The question space is not infinite at the product's chosen resolution (Figure 4). It is 686,116 questions, each answerable ahead of time. The guarantee applies to this fixed corpus, encoder, assignment, and ranking rule. A new image or a changed retrieval rule requires compilation again. Marking a region does not crop and re-encode it: the patch feature retains the full painting's context.

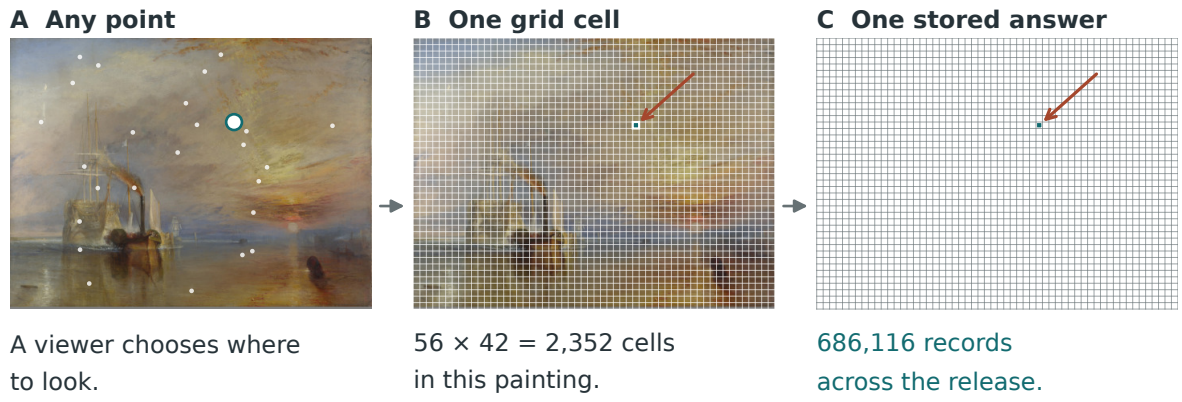


Figure 4: An arbitrary point resolves to one of 2,352 grid cells in this painting and one of 686,116 records in the installed release. The 56 × 42 grid is taken from the release manifest; preprocessing preserves aspect ratio and adds a narrow masked padding strip. The marked point is illustrative. Painting: J. M. W. Turner, The Fighting Temeraire, exhibited 1839, National Gallery, London, NG524; Wikimedia Commons, Public Domain Mark. The image is reproduced for explanation, not selected as evidence of retrieval quality.

4.2 Why DINOv3

I chose DINOv3 because it supplies dense visual features learned without text supervision. That fits the project's aim of finding visual relationships before introducing titles, descriptions, or an authored vocabulary. The encoder is frozen; this project does not train a model from scratch or fine-tune it on the paintings.

DINOv3's training uses self-distillation and objectives that preserve dense feature structure. It brings learned visual regularities to the collection; "without text supervision" does not mean a blank slate. A text-conditioned encoder could support another version of this experience, but it would test a different design choice. No CLIP comparison is claimed here. DINOv3 paper.

Setting	Value	Rationale
Backbone	`facebook/dinov3-vitb16-pretrain-lvd1689m`	ViT-B/16, 768-dim, frozen
Layer	-1 (final, after output norm)	Won a three-way sweep decisively (§9.1)
Long side	896 px, aspect preserved, padded to a multiple of 16	A painting is never stretched to square
Padding	Masked and excluded from the patch set	Padding is not content
Reduction	PCA to 128 components, 79.1% variance	Clustering only; retrieval uses the full 768 dims
Clustering	MiniBatch k-means, k = 200, seed 20260825	Chosen by a band-yield sweep: clean concepts recurring across 30 to 150 paintings, per concept produced. Read with §9.5, which shows the limits of this selection heuristic.

Aspect ratio is preserved without exception. Stretching an image to square changes figure proportions, gesture direction, and compositional balance, which are among the properties under study.

4.3 The metadata wall

Nothing in the ingestion, preprocessing, feature, or discovery path may read artist, title, date, country, school, movement, description, museum, subject, or any human tag. A wall class enforces this in code rather than by convention.

The reason is falsifiability rather than purity. If metadata can reach the feature pipeline, then any apparent discovery may simply be a leak, and no after-the-fact inspection can tell the difference. The wall makes the claim checkable. Metadata joins happen only in retrospective analysis, and only after concepts are frozen.

4.4 What is shipped

File	Size	Contents
`echoes.bin`	13,722,320 B	One 20-byte record per patch: eight destination patch IDs, bit-packed
`concepts.bin`	686,116 B	One byte per patch: its cluster assignment
`manifest.json`	54,512 B	Corpus geometry: image IDs, grid dimensions, offsets, packing widths, release ID, hashes
`labels.json`	4,873 B	One reviewed word per cluster group
`label-descriptions.json`	11,324 B	Per-word `observed`, `reading`, and `coherence`
Binary subtotal	**14,408,436 B (14.41 MB)**	Routes plus concept assignments
All five files	**14,479,145 B (14.48 MB)**	Includes geometry and reviewed text

The release contains no model weights, feature vectors, or pixels. It does contain technical geometry and post-hoc descriptive labels. Those labels do not feed back into feature extraction, clustering, or route ranking. All sizes here use decimal units: 1 MB = 1,000,000 bytes.

5. The compiler

Source: `scripts/compile_echoes.py`. The compiler turns a constrained visual retrieval rule into a fixed-size record for every query patch.

5.1 Problem statement

For each of N = 686,116 patches, find the 8 most similar patches, subject to three conditions.

Exclude the source painting. Otherwise a patch's eight nearest neighbours are its eight immediate spatial neighbours in the same picture, which is trivially true and entirely uninteresting. This one constraint is what makes the result about the collection rather than about a single image.

One result per destination painting. Without it, a single painting holding a large matching region fills the whole wall.

Exactness. No approximate index. The result must be verifiably correct.

An unrestricted pairwise calculation is quadratic: $N(N-1)/2 = 235{,}377{,}239{,}670$ unique unordered pairs, or 941.51 GB if one float32 score per pair were stored. A full N × N matrix would be 1.88 TB. Neither is materialised.

5.2 The cluster restriction

Candidates are restricted to the query patch's own k=200 cluster. Instead of one problem of size N^2, the compiler solves 200 independent problems whose total size is $\sum_i n_i^2$, where $\sum_i n_i = N$. With roughly even cluster sizes (mean ≈ 3,400), cost falls by about two orders of magnitude, and each cluster fits comfortably in memory and in cache.

This is an approximation, and the kind matters. Retrieval within a cluster is exact: every within-cluster pair is scored, which is why the manifest records `candidate_discovery_is_approximate: false` truthfully. What is not guaranteed is that a patch's true global nearest neighbour lies inside its own cluster. A patch near a boundary may have a better match just across it, and the compiler will never see it.

I used the cluster to define a recurring visual neighborhood, then ranked regions within it. This made the retrieval rule inspectable and compilation practical. It also made the partition consequential: a match across a cluster boundary is excluded however high its feature similarity. Section 7.5 now measures this difference against unrestricted DINO retrieval, while Section 8 measures what happens when a new seed moves those boundaries.

Figure 5 separates these two meanings of exactness. Changing the cluster boundary can change the eligible candidates even when feature scores are fixed; this is one route by which seed instability can affect the displayed wall.

An exact ranking can still exclude the best match

Schematic / ten hypothetical candidates from ten different paintings

GLOBAL ORDER	SAME CLUSTER?	STORED RANK
1 A	No · excluded	—
2 B	Yes	1 B
3 C	Yes	2 C
4 D	No · excluded	—
5 E	Yes	3 E
6 F	Yes	4 F
7 G	Yes	5 G
8 H	Yes	6 H
9 I	Yes	7 I
10 J	Yes	8 J

B is exactly first within the pool; A is never considered.
Changing cluster membership can change answers without changing scores.

Letters and order are illustrative, not measured similarities or projected feature coordinates.

Figure 5: Exact scoring does not undo candidate exclusion. In this hypothetical example, A is the global best but falls outside the query cluster, so B is the best eligible destination. Each letter represents a candidate from a different other painting, isolating the cluster restriction from the one-result-per-painting rule. The order is illustrative, not a measurement of this release's recall. No feature projection or invented similarity score is used.

5.3 The inner loop

Within one cluster, for a block of 512 query patches at a time:

Score. L2-normalise all cluster vectors, then perform one dense matrix multiply.

```
scores = vectors[start:end] @ vectors.T        # (512, n) cosine similarities
```

For normalised vectors the inner product is cosine similarity. Handing this to BLAS as a single cache-friendly kernel is the difference between minutes and hours.

Collapse by painting. Patches are sorted by source painting, so each painting's patches form a contiguous run and a segmented maximum reduces the score matrix in one pass.

```
group_scores = np.maximum.reduceat(scores, group_starts, axis=1)
group_scores[np.arange(end - start), group_for_patch[start:end]] = -np.inf
```

The second line masks the query's own painting. Both the one-per-painting rule and the exclude-self rule are enforced structurally, before ranking, rather than filtered afterwards where they would risk returning fewer than eight results.

Select the top eight paintings.

```
top_groups = np.argpartition(group_scores, -keep, axis=1)[:, -keep:]
```

`argpartition` is linear and avoids sorting every painting when only eight matter. Those eight are then sorted by score. Sorting eight entries is a small fixed cost; ties depend on the implementation, so score sorting alone is not a cross-platform determinism guarantee.

Resolve each painting to its best patch. A Numba-compiled kernel scans each selected painting's run and returns its single best patch. The previous step knew the best score; this step recovers the location.

5.4 Bit packing

Patch IDs are packed at the narrowest width that can address the corpus.

```
bits   = ceil(log2(686,116 + 1)) = 20
record = 20 bits x 8 matches = 160 bits = exactly 20 bytes
```

A `uint32` array would need 32 bits per ID, or 21.96 MB. At 20 bits it is 13.72 MB, a saving of 37.5% for arithmetic costing microseconds.

Eight IDs always occupy a whole number of bytes: for any integer bit width b, 8b bits is b bytes. The compiler still checks alignment because a different number of matches need not have this property:

```
if bits * MATCHES % 8:
    raise RuntimeError("The current format requires a whole-byte route record")
```

The all-ones value, 1,048,575, is reserved as a sentinel meaning "no match." Because 2^20 - 1 exceeds N it cannot collide with a real patch ID, and this too is asserted. The sentinel is not decorative. Two clusters in the historical 310-painting release genuinely did not span nine distinct paintings, so rather than padding with invented destinations the compiler records the shortfall and the interface shows a shorter wall.

5.5 Verification

The release is checked at four boundaries: patch geometry, ranking, binary packing, and provenance. These checks make the serving artifact inspectable from source coordinates through to the returned destinations.

Grid validation. Every painting's patches must form a rectangular grid in raster order or compilation aborts. Without this, patch-ID arithmetic in the reader would silently return crops from the wrong offsets.

Exact-search validation. 100 deterministically sampled queries are re-answered by a separately written brute-force routine that shares no code with the compiler. Agreement is reported as top-1 fraction and mean top-8 overlap.

Round-trip verification. 100 random records are read back from the packed binary and compared against the uint32 table before that table is deleted.

Provenance. `run.json` records configuration, seed, repository state, and model identity before compilation. The manifest hashes the feature inputs and route outputs. The new evaluation verifies those input hashes before using the cached features, so it tests the installed artifact rather than a similar run. The remaining requirements for reconstruction from model weights are collected in Section 10.

The Cabinet reader also checks the manifest format, offset-array length, declared byte count, and actual binary length at startup. Truncated files, incomplete copies, and Git-LFS pointers are rejected before serving. These are structural checks; the reader does not recompute SHA-256 on every startup, so same-length corruption requires a separate integrity check.

5.6 Query path

Answering a mark requires coordinate arithmetic, one route-record read, and mapping the returned IDs back to image coordinates (Figure 6). Input coordinates are clamped, including the right and bottom edge, so $x = 1$ or $y = 1$ selects the last column or row rather than stepping outside the grid.

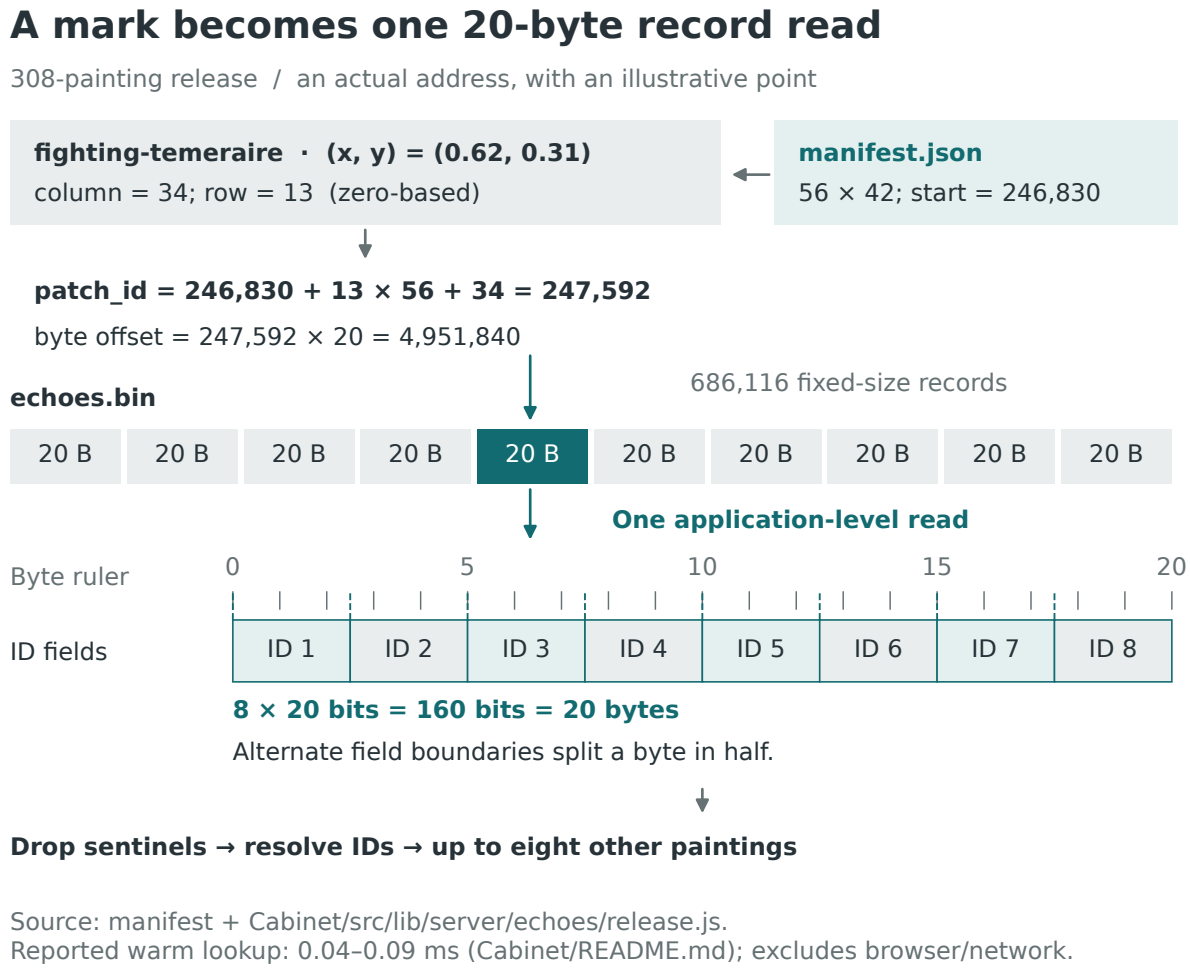


Figure 6: The reader addresses an answer directly. This example uses the actual manifest offset for fighting-temeraire and the illustrative point (0.62, 0.31). Eight 20-bit IDs occupy 20 bytes, saving 37.5% against eight uint32 IDs. The byte ruler shows the half-byte boundaries. Sentinels are removed before IDs become destinations; therefore the result can contain fewer than eight paintings. Source: installed manifest and Cabinet/src/lib/server/echoes/release.js.

There is no nearest-neighbour search or vector arithmetic at request time. Mapping destination IDs uses binary search over the image-offset array. The route binary remains open and is accessed with fixed-size reads; operating-system I/O and caching work in pages or blocks, not literal 20-byte physical transfers. The 686 KB concept table is read once into process memory.

The reader returns coordinates, not pixels, and there is no crop endpoint. The client scales its own already-downloaded painting image inside a clipped window showing only the measured region. Cropping adds no separate crop-image request. A destination painting that is not already cached still has to be downloaded. The crop preserves aspect ratio.

6. Where the compression comes from

The compact artifact comes from storing the answer to the product's question. DINOv3's weights and feature vectors are used offline; the runtime retains the routes, geometry, and reviewed labels. Figure 7 accounts for the storage at each representation, using the **installed 308-painting release**.

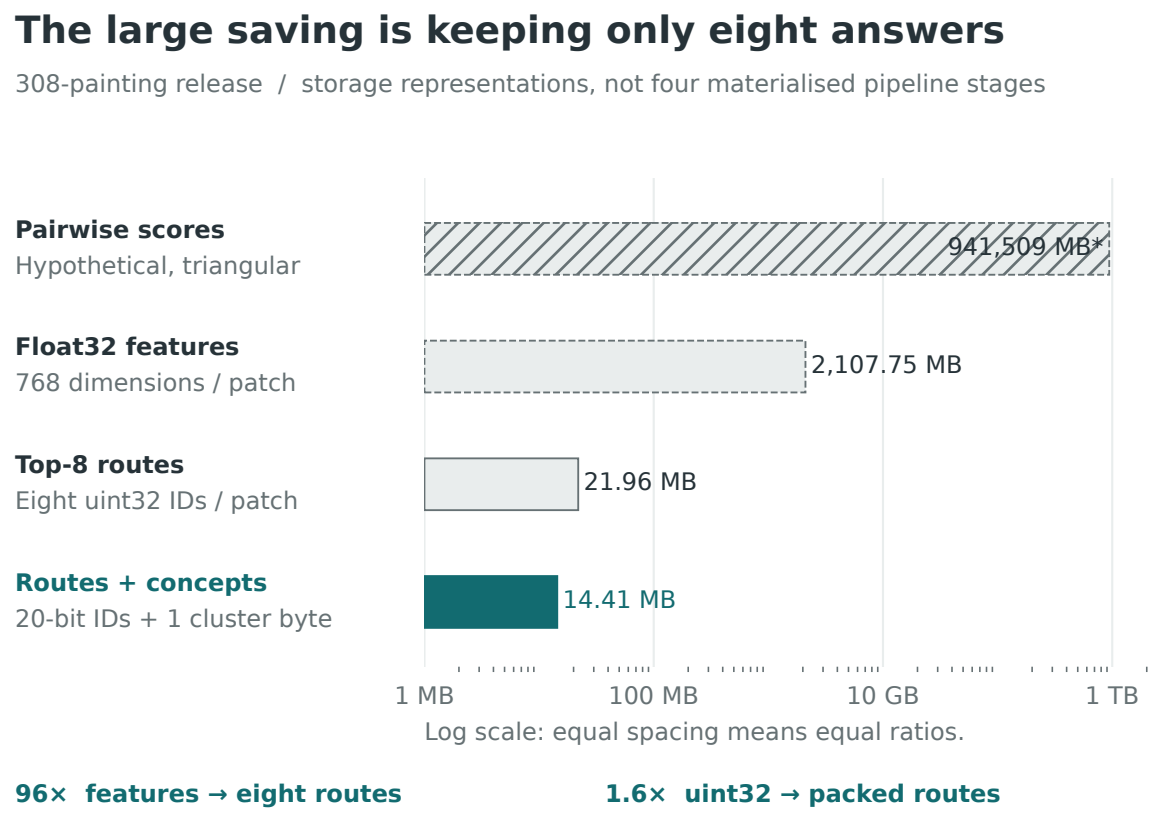


Figure 7: Restricting the answer to eight destinations reduces storage from 2,107.75 MB of float32 features to 21.96 MB of uint32 routes, exactly 96×. Packing the routes to 20 bits per ID contributes a further 1.6× reduction, or 37.5%. The final bar adds the 0.69 MB concept table, giving 14.41 MB; all five release files total 14.48 MB. The 941.51 GB triangular pairwise store is hypothetical and was never built. Source: installed release manifest and explicit byte-count formulas in the figure renderer.

The features-to-routes reduction dominates, and its logic is a product judgement rather than a mathematical one:

> A viewer will look at eight results. Storing the ability to produce any other answer is storing something nobody will ever ask for.

Similarity scores go for the same reason. The wall shows eight regions in rank order and displays no percentages. A similarity number is not comparable across models or queries, and showing one would imply a precision the measurement does not have. Rank order is the only part of the score that reaches the interface, and rank order is already preserved by storage order. Keeping the scores would add bytes without supporting another interface operation.

The last step is ordinary bit packing, saving 37.5% of route bytes. Adding the concept table makes the final binary subtotal 1.52× smaller than uint32 routes alone; that ratio and the routes-only 1.6× ratio have different denominators.

What the size comparison tells us. The two binaries are 146× smaller than the raw float32 feature cache. That is a useful accounting of this pipeline's reduction, rather than a benchmark against the best possible search index. A product-quantised representation at 32 bytes per patch would use roughly 22 MB before index overhead, placing it in the same broad size range. No ANN index or matched ANN recall benchmark was built for this paper.

The reason to choose the compiled table is its fit to a closed interaction: answers are already ranked, record size is fixed, and request-time code needs neither a vector index nor vector arithmetic. A search index remains a useful alternative when the corpus, query set, or retrieval rules need to change without rebuilding every answer. Section 7.5 measures global DINO agreement, which is a separate question from ANN performance.

How size grows. With eight destinations and a bit width $b = \lceil \log_2(N + 1) \rceil$, routes occupy Nb bytes and the one-byte concept table occupies another N. Storage therefore grows linearly while the bit width is fixed, with small steps when it increases. This is separate from the within-cluster quadratic work needed to compile the answers. The shipped file does not grow like the full pairwise relationship matrix.

Why not smaller? The obvious next step is delta-coding spatially adjacent patches, which tend to route to similar destinations. It was not pursued because 14 MB already sits below the threshold where size affects the product.

The route binary is bundled with the server function, not downloaded once per visitor. Therefore dividing a hosting transfer allowance by 14.41 MB does not estimate visitor capacity for this architecture. Browser traffic consists of API responses and application assets, with paintings served separately from Commons. A meaningful cost model would measure those bytes, request volume, function time, and the hosting plan actually used.

The operating benefit is **no model-inference fee per query**. Lookup still consumes CPU, memory, storage I/O, and network resources. Feature extraction and recompilation also cost work whenever the corpus or retrieval rules change.

7. Results

7.1 Compiler correctness

The compiled release is exact within its candidate pool, and this was verified rather than assumed. Across 100 deterministically sampled queries re-answered by an independently written brute-force search:

Metric	Value
Exact top-1 agreement	1.000
Mean top-8 set overlap	1.000
Minimum top-8 set overlap	1.000

Every sampled query agreed perfectly, which exercises the blocked matrix multiply, the segmented maximum, the painting-collapse logic, and the bit packing end to end.

This revision adds a second check: 1,000 uniformly sampled query patches from the exact shipped feature cache were ranked again and compared with their stored answers. All destination sets agreed. The original 100-query test also checked top-1 order with an independently written brute-force routine. These are strong checks of the implementation's stated contract: exact ranking within the assigned candidate pool, one result per other painting, and correct packing. Visual relevance is evaluated separately from agreement with that contract.

7.2 Coverage

In the September 2026 release, 924 local API queries probed two opposite corners and the centre of every painting.

Metric	Value
Average results returned	7.99 / 8
Empty walls	0
Duplicate destinations	0
Unopenable results	0

The shortfall from 8.00 is one sparse cluster spanning only three paintings, whose 975 source patches can return at most two destinations. The compiler records that fact rather than inventing destinations to fill the gap.

For this edition, an exhaustive audit of the stored binary counted 685,141 records with eight destinations (99.86%) and 975 with two (0.14%), with no other cardinalities. This is a census of stored answers, separate from the 924-query API sample above; it does not retest image availability or HTTP behavior. Figure 8 shows the first underfilled record in ascending patch-ID order. A short answer reflects the number of eligible paintings, not a novelty threshold or the noise label that the research pipeline still lacks.

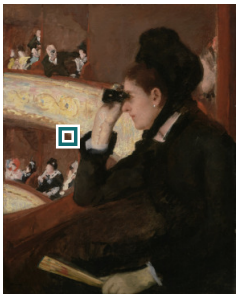
Two destinations is the complete answer

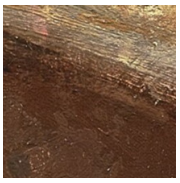
308-painting release / first underfilled patch in ascending ID order

QUERY · Burne-Jones

Patch 110221

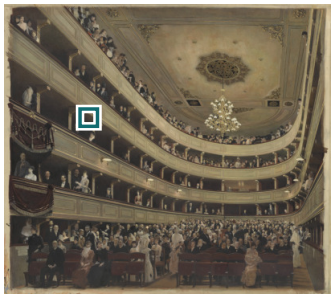
1 · Cassatt

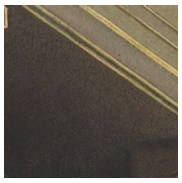




Patch 134265

2 · Klimt





Patch 348820

EIGHT STORED FIELDS

134265	348820	empty	empty	empty	empty	empty	empty

Six sentinel fields are removed. No invented or duplicate destinations.

FULL RELEASE CENSUS

685,141 records return eight (99.86%); 975 return two (0.14%).

Three paintings in this cluster permit two other paintings. This is not a novelty/noise test.

Figure 8: Returning two is the faithful answer for patch 110221. Its cluster spans three paintings, leaving only two other paintings eligible: patches 134265 and 348820, shown in stored order. The remaining six fields contain the sentinel 1048575, displayed here as empty and removed by the reader. Full paintings and enlarged 3 × 3-cell neighborhoods preserve aspect ratio; credits are in Appendix A. The census counts every record in echoes.bin. Neither the plate nor the count establishes retrieval quality or semantic novelty.

7.3 Performance

The Cabinet README reports the following timings on a built local server:

Metric	Reported value
Warm lookup per mark	0.04 to 0.09 ms
Cold initialization, including both binaries	~45 ms
Memory growth over thousands of lookups	Reported flat
Model invocations per query	0

These are reported project measurements, not a new benchmark in this paper. A raw timing trace, hardware description, percentile distribution, and matched hosted-model benchmark are not supplied. The numbers describe the local lookup, not HTTP round-trip latency or the time to display eight paintings. Network latency, function startup, browser rendering, and destination-image downloads remain outside that timing. The current review report explicitly records local validation and no live-deployment verification.

7.4 What the concepts look like

On the **1,005-painting research corpus**, k = 200 yields 129 clean concepts and 71 flagged as artifacts. That is the corpus used for the backbone comparison in Section 9.2; the earlier layer and k sweeps used 310 paintings. The installed 308-painting release is clustered separately.

A reviewer found recurring faces, hands, and held objects across paintings, alongside vertical bundled forms such as folds, columns, and sheaves. Necklines, the collar-and-shoulder junction, offered a particularly useful example of a visual regularity that is easier to show than to name. No such descriptions were supplied to the discovery pipeline.

Three artifact clusters closely follow image corners, with edge shares between 0.70 and 0.79. They point to frames, mounts, and scan margins: reproduction features that recur alongside painted content. Flagging and demoting these groups makes that signal visible in the review.

These contact-sheet readings helped choose a useful configuration. They are observations by one reviewer, rather than a blinded relevance study. Descriptions such as hands or necklines identify what the reviewer saw in a particular run; they do not turn a cluster number into a persistent identity. Section 8 measures that distinction on both the historical snapshot and the installed release.

7.5 Candidate restriction and low-cost controls

This revision compares retrieval rules on the **same 1,000 uniformly sampled patches from the installed 308-painting release**. An unrestricted DINO reference searches all 686,116 content patches, excludes the query painting, and returns the best region from each of the eight highest-ranked other paintings. Compared with that reference, the shipped cluster-restricted answers retain **54.35% of exact destination patches** and **74.04% of destination paintings** on average; top-1 patch agreement is **61.7%**. This measures the effect of the candidate-pool rule directly. It is neither an ANN benchmark nor a relevance score.

I also ran three low-cost controls over those same content cells: a 48-D RGB histogram (16 bins per channel), 48-D pixel features (4 × 4 RGB area averages), and a 36-D HOG descriptor (a 16 × 16 window, 8 × 8 cells, nine orientation bins, L2-Hys). Each uses cosine ranking over all patches, the same painting exclusion, and one result per destination painting. None inherits DINO's clusters.

Local descriptor	Exact patch retention	Painting retention
RGB histogram	0.0000%	4.55%
4×4 RGB pixels	0.0000%	3.74%
HOG	0.0125%	3.60%

The table reports mean overlap with the **unrestricted DINO reference**, not the shipped restricted answer. Only one of its 8,000 destination patches is also returned by HOG; none is returned by the other two controls. No control shares the reference's first-ranked patch on these queries.

The useful conclusion is specific: these local colour, pixel, and gradient statistics do not reproduce DINO's routes under this protocol. It does **not** follow that DINO's routes are better, or that a transformer is necessary. DINO's patch vectors include surrounding image context, whereas these controls describe a single 16×16 cell. A matched-context comparison and blinded relevance review would test perceptual value. This first control measures retrieval agreement; it does not rerun the concept-discovery experiment with handcrafted features.

The query IDs, per-query overlaps, configuration, library versions, input hashes, and alternate projectors are recorded by `evaluate_release.py`. Appendix A provides the reproduction entry points. The sample is uniform over patches, so paintings contribute in proportion to their content-cell count.

8. Reproducibility at two levels

There are two useful questions here: does an installed release answer the same point consistently, and does a new clustering preserve the same relationships? The first is a property of the compiled artifact. The second needs measurement. I tested them separately so the product could have a clear release boundary while the research continued toward more stable concepts.

8.1 What the historical experiment established

The original **310-painting snapshot** held features, k, PCA, and preprocessing fixed while varying the k-means seed across four alternatives. For each baseline concept, member retention is the fraction of its patches landing in the largest destination fragment under a new seed. Matching is many-to-one: it measures how much stays together, without penalising unrelated patches joining that group. It is not Jaccard overlap or one-to-one concept recovery.

A concept qualifies as durable only if it retains at least half its members under **every** tested seed. Of 118 clean baseline concepts, 9 qualified (7.6%). The median of each concept's worst-seed retention was 0.337; the p10 and p90 were 0.20 and 0.49. Taking the minimum first prevents three good runs from hiding one that breaks a group apart.

The useful lesson was to test the kind of stability the next design would need. The group reviewed as faces retained 0.64 across a k sweep but only 0.31 across seeds; the group reviewed as elbows retained 0.18 across seeds. Hands and necklines retained 0.65 and 0.51. Those observations guided further inspection, while the seed test showed why numbered clusters should remain provisional.

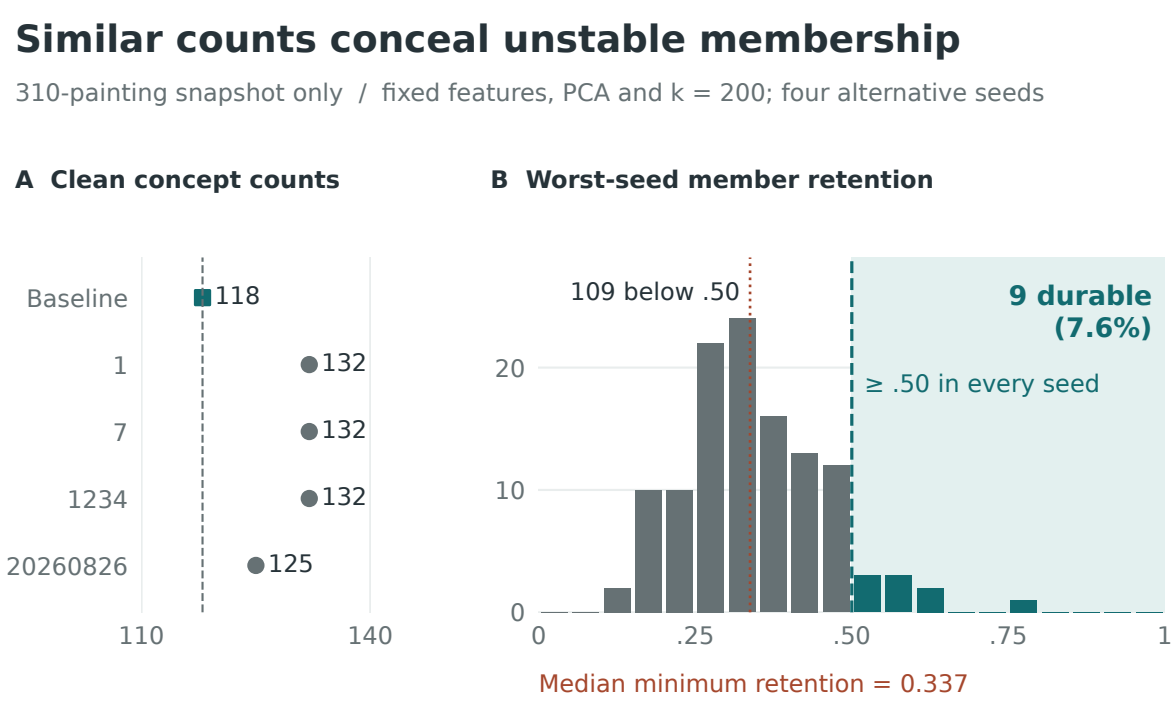


Figure 9: Historical 310-painting snapshot: similar counts do not imply stable membership. Of 118 baseline clean concepts, 9 (7.6%) retain at least half their members in their largest destination fragment under every tested seed. The histogram shows minimum retention across four seeds, with median 0.337. Reviewer descriptions identify four watched groups in that experiment, not persistent labels. Source: analysis/stability/seed_stability.json.

8.2 Repeating the test on the shipped corpus

For this revision, I repeated the retention measurement on the **installed 308-painting release, with 686,116 patches and 131 clean baseline concepts**. The evaluation verifies the frozen input hashes, reuses the installed 128-D PCA, and holds the original 600,000-row fitting sample fixed. Only the k-means seed varies: 1, 7, 1234, and 20260826 against baseline 20260825, at k = 200. A same-seed control first reproduced every installed assignment.

18 of 131 clean concepts (13.7%) retain at least half their members under all four alternate seeds. Median worst-seed retention is **0.333**; the p10 and p90 are **0.195 and 0.538**. The result includes every clean baseline concept, with no selection by its visual description.

This confirms that concept identity still needs work on the corpus visitors actually use. The historical and installed results belong to separate runs and fitting protocols; their difference is not a controlled estimate of the effect of removing two paintings. The release-specific measurement replaces the need to infer today's behavior from the older snapshot.

8.3 What changes when the release is rebuilt

The same evaluation sampled **1,000 patch IDs uniformly without replacement**, using a fixed seed chosen before inspecting the results. Each query was ranked again under every alternate clustering, using the installed DINO features, cosine ranking, exclusion of the source painting, and at most one region per other painting. These sampled answers are exact within each new candidate pool; there is no need to compile the other 685,116 records to measure this sample.

Wall retention is the fraction of a stored answer retained after reseeding. I report both exact patch IDs and destination painting IDs: the same painting can remain on a wall while its highlighted region changes. The metric measures set membership, not rank order or human relevance.

Across the four seeds, mean retention is **44.7% of exact patches** and **70.7% of destination paintings**. Figure 10 shows each seed separately. These figures describe the chosen 1,000-query sample, not the full corpus of possible answers. They also separate two visible changes: a wall can keep a painting while moving its crop, or replace that painting entirely.

This gives the architecture a concrete contract. A frozen release returns the same stored routes, while rebuilding is a change to the experience that deserves review. Fixed feature vectors preserve the scoring rule; they do not preserve the cluster-defined candidate pool. The measurements show the size of that change without assuming that either a retained or a replacement result is better.

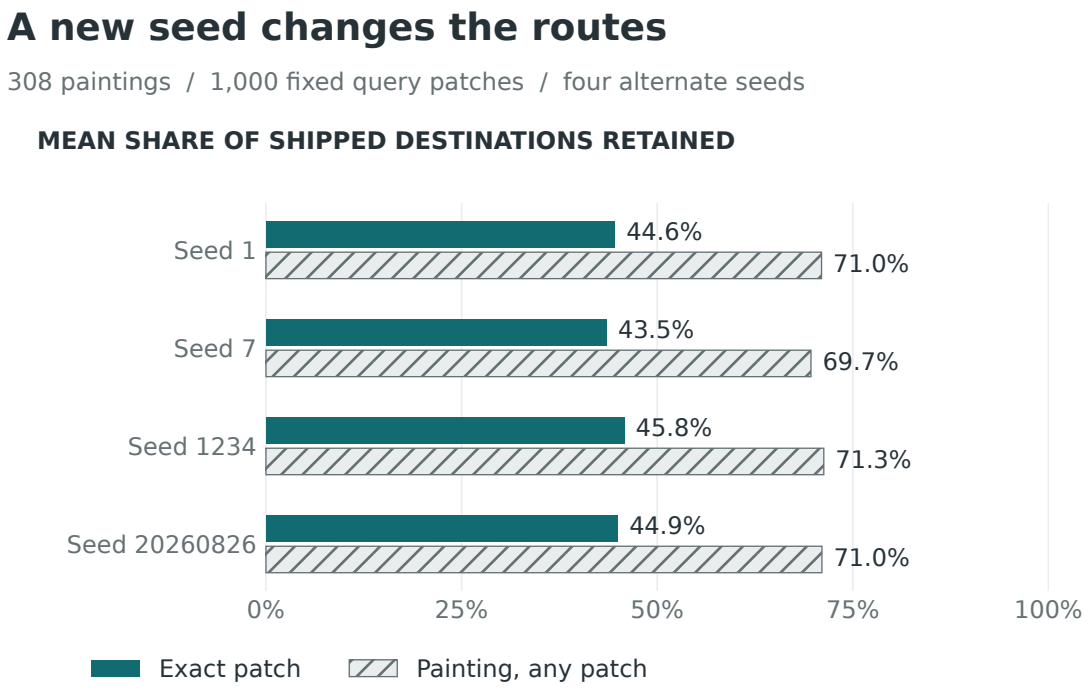


Figure 10: Installed 308-painting release: changing the seed changes routes. Each pair of bars summarizes the same 1,000 uniformly sampled queries, compared with their shipped answers. A painting counts as retained even if its selected patch moves; exact-patch retention requires the same patch ID. Features, PCA, fitting rows, and retrieval rules are fixed. These are mean set-retention measurements, not relevance scores or a census of all possible queries. Source: evaluation/results.json; method and seeds in evaluation-config.yaml.

8.4 Keeping interpretation attached to a release

The implementation treats cluster numbers as internal positions in one run. The API returns a reviewed word and a provisional flag; saved findings store the word rather than the raw cluster ID. Installing a new release disables the old concept names until its groups are reviewed again. Historical words remain readable for saved findings, but are not automatically attached to new clusters.

For the installed release, the review record is `assistant_visual_review`: an assistant inspected the first 16 crops per cluster after discovery, at my request. This is an editorial layer over text-free features, with no independent human rating. Adjacent patches and repeated paintings also mean the 3,200 crops are not independent observations.

Of the 29 active words, 7 are rated *tight*, 20 *mixed*, and 2 *loose*. The loose labels explain their uncertainty and use a dashed rim in the interface. The 40-entry description file additionally retains 11 historical words for saved findings; its totals of 16 tight, 22 mixed, and 2 loose describe that wider file, not the active vocabulary.

These choices make the release boundary explicit. They preserve readable saved findings and prevent a new clustering from silently inheriting old names. Establishing that the words remain semantically useful requires a separate review, just as judging changed routes requires looking at the returned images.

9. What the experiments taught me

The experiments below made the final design simpler. They helped me choose which settings to retain, which possibilities to set aside, and which questions needed a different test. Corpus sizes are stated locally because each result belongs to a particular stage of the project.

9.1 Middle layers are worse than the final layer

Prevailing intuition, and the project's own plan, held that middle transformer blocks carry part-level structure and would sharpen concepts such as necklines and limb junctions. Layers -1, -4, and -8 were extracted and clustered identically on the 310-painting corpus.

	L-1	L-4	L-8
PCA variance at 128 components	0.791	0.929	0.995
Clean concepts	118	109	19
Flagged as artifact	82	91	181
Band yield	0.47	0.06	0.03

The variance row helps interpret the difference. At layer -8, 128 components absorb 99.5% of the variance, so the mid-stack representation is nearly low-rank and k-means is subdividing a much smaller effective space. Ninety percent of the result is position and texture rather than parts. That comparison supported keeping the final layer for this collection.

A correctness bug surfaced during this sweep and is worth recording. Raw intermediate activations grow sharply in scale with depth, with standard deviations of 3.58 at layer -8 against 0.40 at layer -1 after normalisation, a 162x spread in raw form. Comparing them directly would have made layers differ by magnitude rather than by content, and PCA and k-means would have faithfully reported a difference that was pure artifact. Applying the model's output norm at every layer cut the spread to 8.9x. The practical lesson is to match the normalisation as well as the layer index before interpreting a representation comparison.

9.2 A bigger backbone made concepts broader, not sharper

DINOv3 ViT-L/16, at 300M parameters and 1024 dimensions, was run over the 1,005-painting corpus with layer, k, seed, and PCA held fixed.

	ViT-B/16	ViT-L/16
Clean concepts	129	130
Band yield	0.07	0.04
Median paintings per concept	361	411

The larger backbone produced broader clusters under these settings and was not adopted, at the reported 2.2x extraction cost. Feature sharpness itself was not measured.

Band yield here is far below the 0.47 in Section 9.1 because these runs use the 1,005-painting corpus, where a fixed band of 30 to 150 paintings is a much narrower target than it was at 310. The figures are comparable within this table, which is what the decision rested on, and not across corpus sizes.

One confound belongs with the result. Feeding 1024 dimensions into the same inherited 128 PCA components drops retained variance to 76.8%, so ViT-L ran handicapped by a setting tuned for a 768-dimensional model. ViT-L at roughly 170 components is untested and cheap,

since the embeddings are cached. The result supports the smaller backbone under the tested settings; a comparison at matched retained variance remains a useful follow-up.

9.3 Keeping k-means after testing HDBSCAN

On the historical 310-painting corpus, I tried HDBSCAN because it can leave a patch unassigned and does not impose k-means' centroid-based partition. That made it a useful candidate for the missing noise label.

At the specified 128 components it finds 2 concepts. Distance concentration is a plausible explanation, but this run did not isolate that mechanism. Two output clusters show failure under the tested settings, not proof that the features contain only two structures.

Producing concepts at all required dropping to 32 components, discarding 22 percentage points of variance that k-means was using successfully. Its headline 89% noise fraction first looked like a substantive finding, as though most of a painting were not recurring structure. It is more likely an artifact of density estimation in a space missing 43% of its variance. On visual inspection of the contact sheets, k-means is clearly better.

The noise-label objection remains open and legitimate. k-means still cannot say that a region belongs to no concept. The tested HDBSCAN settings did not provide a useful replacement here.

9.4 A single extraction scale was enough for this approach

In the historical 310-painting experiments, pooling features at 224, 448, and 896 pixels lowered band yield to 0.420 against 0.505 for 896 alone. Concepts that appeared scale-pure turned out to follow whichever scale was oversampled. The features are near-collinear across scales, with cosine similarities between 0.94 and 0.99, so the extra views add cost without adding information.

9.5 Use metrics to narrow the search, then inspect the result

Band yield rated HDBSCAN at 0.44 against k-means at 0.47, nearly a tie, on contact sheets that were visibly and substantially worse. Silhouette, elbow, and similar internal scores were likewise uninformative here.

The lesson was to use a score for the comparison it can support. In the 310-painting k sweep, band yield rose from 0.10 to 0.30 to 0.47 to 0.49 at $k = 50$, 100, 200, and 400. The improvement flattened at 200, while concepts appearing in fewer than 20 paintings jumped from 7 to 45 at $k = 400$. Together with the contact sheets, that supported $k = 200$ as a practical choice.

A near-tie between different clustering methods did not carry the same evidence. The large gaps helped order experiments; close scores required looking at what the regions actually contained. Independent, blinded review would make those visual judgments more transferable. It is a different evaluation task from checking the compiler's arithmetic.

9.6 Keeping the visitor in control of the point

A tempting optimisation: scan the concept table at build time, work out where each vocabulary word is strongest in each painting, and ship that as a table of findable points.

I built this version and then set it aside. It costs 23 KB gzipped on first paint, it hard-codes an answer the release already gives for free, and, decisively, it converts "mark what catches your eye and be told what it is" into "click the spots we picked." That reintroduced the constraint I had wanted to remove from the hand-authored design. The compiled lookup already answers

any point with one 20-byte read, so the extra table did not earn its cost or its effect on the interaction.

9.7 Distance alone was not a useful novelty detector

Because k-means assigns every patch to its nearest centroid however far away that centroid is, the system cannot honestly report that a region belongs to nothing. The historical 310-painting calibration tested centroid distance as a proxy: pure noise reads at only 1.06x the median corpus distance. The ArtML research viewer labels it as weak. This is a separate interface from Cabinet, whose compiled route table contains neither these distances nor a calibrated novelty score.

10. Scope and remaining questions

The measurements support a clear systems result: the fixed interaction can be compiled into a small, verified serving artifact. The boundaries below identify what the next experiments should address.

Candidate selection changes the answer. The compiler is exact within its assigned cluster. On the 1,000-query sample, its answers retain 54.35% of the exact destination patches and 74.04% of the paintings from unrestricted DINO retrieval (§7.5). The restriction is part of the product's retrieval rule, rather than an approximation error in the compiler.

A rebuild is a new release. Both the historical and installed seed tests find unstable concept membership. Section 8 now measures the effect on sampled walls directly. It does not establish whether changed results are better, worse, or equally useful. Consensus clustering remains a candidate for more stable groups, not a demonstrated solution.

Relevance needs its own evaluation. The new colour, pixel, and HOG controls produce different routes from DINO (§7.5). That difference does not show which answers people prefer or prove that a transformer is necessary. These are local cell descriptors; DINO features include surrounding image context. Matched context controls, clustering comparisons, and blinded relevance judgments remain open. CLIP was not tested: excluding text-conditioned features expresses the project's discovery objective, not measured superiority over CLIP.

Interpretation follows discovery. The current words are assistant-authored review labels, with no independent human rater or inter-rater agreement (§8.4). Visual resemblance alone establishes neither influence nor shared meaning. Likewise, the preference for an experience built around looking is my design intention; no visitor study has yet measured enjoyment, usefulness, or return.

Coverage is different from abstention. Every patch belongs to a k-means cluster. The project still needs a defensible way to say that a region belongs to no recurring concept. A short stored answer only means too few eligible paintings, not a calibrated novelty or confidence judgment. Frames and scan margins remain in the feature space, flagged and demoted rather than removed.

The evidence describes this collection. The installed corpus contains 308 European and American paintings, not a representative sample of art. The expanded 1,005-painting experiments and historical 310-painting experiments are identified where used; they are not pooled into one benchmark.

Work moves offline. There is no model invocation per lookup, but extraction, clustering, and compilation still consume resources. The paper does not provide an end-to-end cost or energy trace. Compilation is quadratic within clusters; its rebuild cost needs remeasurement as the

collection grows. Reported local lookup timings exclude HTTP, browser rendering, and image downloads.

The artifact has explicit boundaries. Catalogue and release IDs must match and are deployed together. More than 256 concept IDs would require a wider concept table. Route records grow when the patch-ID bit width increases; eight IDs remain byte-aligned, while other match counts may need padding or a revised format. Structural startup checks reject truncation but do not detect all same-length corruption.

Cached-artifact reproduction is stronger than model reconstruction. The installed manifest supplies input and output hashes, which the new evaluation checks. The original run does not pin an immutable model revision and weight-file digest. Reproducing the exact cached-feature experiments is therefore better specified than recreating that feature cache independently from downloaded weights. Closing this provenance gap would strengthen future releases.

11. Future work

11.1 Consensus clustering

Cluster across several seeds and keep what co-occurs, rather than trusting any single partition. This is the primary candidate fix and it addresses two problems at once: stability, by defining a concept as agreement across runs rather than as a position in one run; and the missing noise label, since a patch that lands with different companions under every seed is precisely a patch belonging to no stable concept.

It can operate in the existing 128-component feature space, but need not preserve existing concept boundaries or IDs. Stable groups and an abstention rule must be defined and evaluated; consensus does not guarantee either useful stability or a calibrated noise label. It is the next candidate, not a demonstrated fix.

11.2 Review the changes that the seed audit found

The two direct measurements are now complete on the shipped corpus: member retention and sampled wall overlap (§8). The next step is to inspect changes, especially queries with low retention and patches near cluster boundaries. A blinded comparison of old and new walls can distinguish a harmless change of example from a loss of useful resemblance. Repeating the fixed sample for future releases would make that review comparable over time.

11.3 Compare relevance as well as route agreement

The three low-cost retrieval controls are now measured (§7.5). Extend them with larger context windows and matched clustering protocols, then ask independent reviewers to compare the returned regions without knowing their source method. This would test whether DINO's different routes offer a useful perceptual gain, rather than treating disagreement as success.

A CLIP image-feature control could separately examine how text-conditioned pretraining changes the discovered structure. That is a comparison of design choices, not a prerequisite for the finite-query compilation result.

11.4 ViT-L at matched retained variance

Re-run the large backbone at roughly 170 PCA components so retained variance matches the baseline's 79.1%. The embeddings are cached, so this is inexpensive, and it closes the confound in Section 9.2 honestly in either direction.

11.5 Spatial grouping over the label grid

Pooling scales did not improve this approach (§9.4), while object-scale structure remains desirable. Grouping connected regions that share a concept over the existing label grid is the cheap route to it and needs no re-extraction.

11.6 Delta-coding the route table

Spatially adjacent patches route to similar destinations, so delta-coding neighbours would likely cut the binary substantially. Deliberately deferred: Section 6 argues 14 MB is not a problem worth optimising.

11.7 On-device inference

On-device foundation models could in principle run the original interpretive design locally without a hosted inference fee, recovering design one's output without design one's economics. Not pursued, because it is currently platform-specific, so it would serve a fraction of visitors while requiring a second full implementation. Worth revisiting when equivalent capability reaches the browser.

11.8 Other domains

The compiler is domain-agnostic. It needs only a frozen patch-level encoder, a fixed corpus, and a fixed answer count. Any collection where "show me where else this appears" is the interaction fits the same shape, including archives, scanned manuscripts, and microscopy.

Transferring that architecture also requires a suitable backbone. DINOv3 is trained on natural images, which is why it transfers to paintings at all. A domain whose relevant structure is invisible to a natural-image model, such as hairline fractures in machined parts, would need its own encoder first, and that is the expensive part the current design avoids.

11.9 Adaptive representations

Adapting the backbone to this corpus remains gated on stability, deliberately. Without a stable clustering, an adaptation result cannot be told apart from reseeding noise. The larger frozen-backbone experiment (§9.2) provided a lower-cost comparison before considering adaptation.

12. Related work

Self-supervised visual representation learning. DINOv3 provides the frozen dense features used here. Its contribution includes preserving dense structure through Gram anchoring, alongside large-scale self-supervised training. Cabinet contributes neither a new encoder nor a new training objective; it compiles a fixed interaction over those features. Simeoni et al., DINOv3.

Compact nearest-neighbour search. Product quantisation compresses vectors so similarities can be estimated from compact codes. It is a relevant alternative when queries are not enumerated in advance. The 32-byte-code example in Section 6 is size arithmetic, not an implemented competitor or a recall benchmark. Jégou, Douze and Schmid, Product Quantization for Nearest Neighbor Search.

Visual correspondence in art collections. Patch and region retrieval without curatorial text predates this project. Shen, Efros and Aubry use spatially consistent feature learning and geometric verification to find repeated patterns in artworks. Their work provides context

for visual matching without curatorial text. Cabinet focuses on a different systems question: compiling a closed query set into a small serving artifact and accounting explicitly for what can be discarded. Discovering Visual Patterns in Art Collections.

Materialised views and precomputation. The compiled table can be understood as a materialised answer set over a finite query domain. It trades flexibility for a bounded lookup: new paintings, a different encoder, or different ranking constraints require rebuilding. No novelty is claimed for precomputation or bit packing themselves.

Model compression. Distillation, quantisation, and pruning aim to retain a model's ability to process inputs. This release retains answers only. It cannot embed an unseen painting or answer a new kind of query.

13. Conclusion

I started with a simple intention: let someone follow a detail through an art collection. The three designs each contributed to the result. The model-backed prototype established the interaction; hand authoring exposed the value and the limits of preparing regions in advance; visual encoding made complete coverage practical. Compilation then moved the expensive work out of the visit itself.

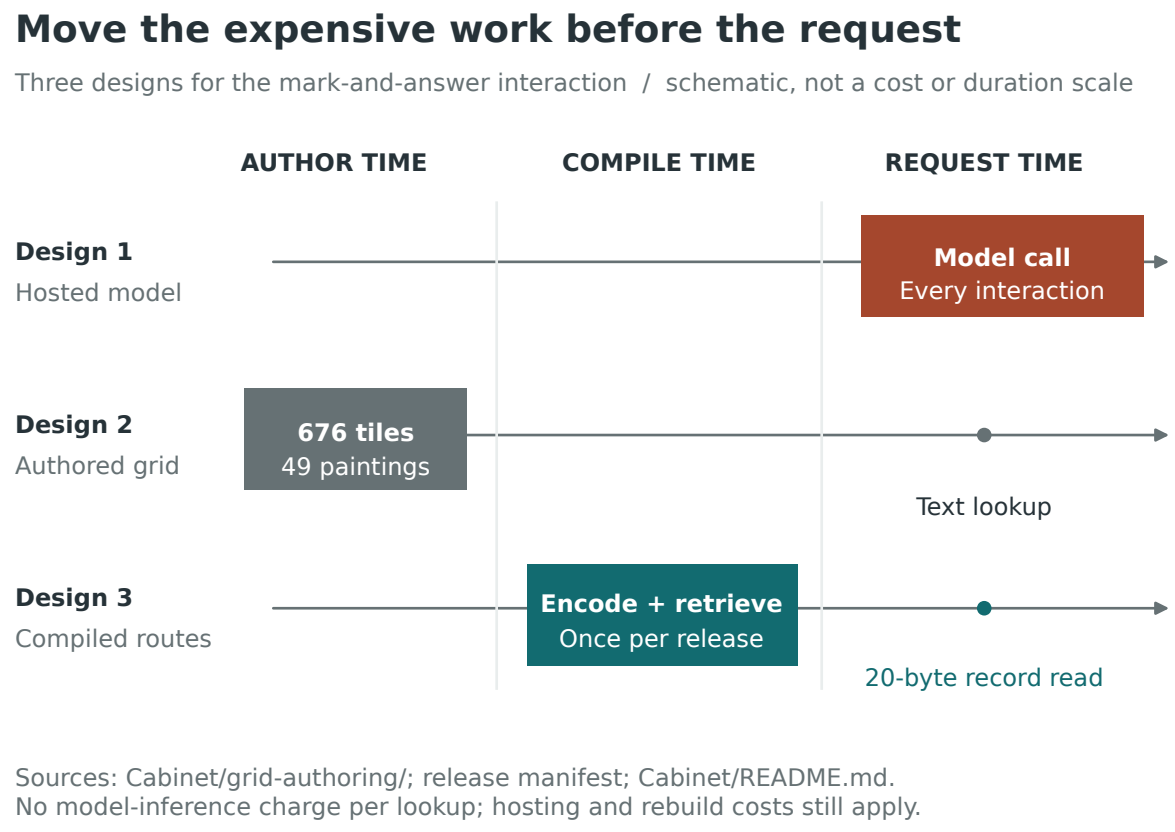


Figure 11: Compiling the fixed interaction moves feature extraction and retrieval before the request. The authoring archive contains 678 PNG artifacts across 49 paintings, including 676 numbered tiles and two auxiliary images. The compiled release answers all 686,116 patch addresses with a 20-byte route record each. Lane widths are schematic, not measured time or money. No model-inference fee is incurred by a lookup; hosting and rebuild work still have a cost. Sources: Cabinet/grid-authoring/, the installed manifest, and Cabinet/README.md.

The key step was recognising the shape of the question. A person can mark any point, but at the chosen resolution every mark has an address. Once those addresses are enumerable, the system can compute their answers once and retain only what the interface needs. In this release, that means 686,116 records, up to eight other paintings per record, and a complete artifact of 14.48 MB.

The new measurements make the design boundary concrete. Exact compilation is verified; seed changes can alter the returned regions, so the release is the unit of reproducibility. The low-cost controls provide another reference point for the feature choice, with human relevance still a separate question. These findings guide the next iteration without changing the finite-query insight.

That is the ghost in the gallery: the model's visual work remains in the routes, even though the model is absent when the visitor arrives. A mark brings back other places to look. The visitor supplies the curiosity; the runtime reads an answer already waiting for it.

Appendix A: reproduction

Artifacts

Path	Contents
`ArtMLModel/scripts/compile_echoes.py`	The compiler
`ArtMLModel/src/artvision/app/echoes_compiled.py`	Packing format and reference reader
`ArtMLModel/integrations/vercel/visual-echoes.ts`	Production reader
`ArtMLModel/configs/experiments/mvp.yaml`	Frozen baseline configuration
`ArtMLModel/docs/decisions/gate-m.md`	Sweeps, stability, negative results
`cabinet/docs/ECHOES-PIPELINE.md`	Rebuild procedure
`cabinet/docs/ECHO-LABELS.md`	The two-vocabularies measurement
`cabinet/grid-authoring/`	Design-two artifacts, 676 numbered tiles plus two auxiliary PNGs
`cabinet/src/lib/ai/prompts.js`	Design-one prompt, preserved

Release identity

```
Release   2dfcd0e5e197a8a9-k200-420b031f81de
Format    artvision-echoes-v1
Space     k200, layer -1
Paintings 308
Patches   686,116
Packing   20 bits x 8 = 20 bytes per record
Sentinel  1,048,575
```

The manifest records SHA-256 digests for the binary artifacts. Compilation and installation can check those digests; the Cabinet runtime checks lengths and structure, not hashes on every startup (§5.5). The recorded compilation time is 9 September 2026, 16:24:20 UTC. See §5.5 for provenance limitations.

Rebuilding this paper and its figures

Markdown is the editorial source. The LaTeX version is generated from it so the figures, qualifications, and appendices cannot silently diverge. From the project root, with the project's Python environment and Pandoc installed:

```
.venv/bin/python docs/whitepaper/render_figures.py
.venv/bin/python docs/whitepaper/build_paper.py
```

The renderer uses only the included figure-data snapshot. PDF figure assets are embedded in LaTeX; SVG assets are referenced in Markdown. See figures.md for the optional evidence refresh and the PDF compilation command.

Artwork credits for Figures 1, 2, 4, and 8

The following credits follow Cabinet's existing artwork-research records, which identify these images as public-domain reproductions on Wikimedia Commons. Titles and artists are publication credits, not retrieval inputs. The plate preparation verifies original image hashes against the release's approved-image record before making thumbnails and crops. No original is modified, no image is stretched, and no detail is sharpened or reconstructed. The enlarged details are display crops of the original reproductions; pixelation and surface texture are retained. Figure 2 uses the same query point as Figures 4 and 6.

Figure 1, query and destinations in stored order:

- **Query:** Johannes Vermeer, *Girl with a Pearl Earring*, circa 1665, Mauritshuis, The Hague. Image record.
- **1:** Antonello da Messina, *Portrait of a Man*, about 1475–1476, National Gallery, London, NG1141. Image record.
- **2:** Raphael, *La Fornarina*, about 1520, Palazzo Barberini, Rome, inventory 2333. Image record.
- **3:** Jean-Baptiste Camille Corot, *Woman with a Pearl*, Louvre, Paris, RF 2040. Image record.
- **4:** Lucas Cranach the Elder, *Judith with the Head of Holofernes*, around 1530, Kunsthistorisches Museum, Vienna, GG 858. Image record.
- **5:** Giovanni Bellini, *Portrait of Doge Leonardo Loredan*, about 1501–1502, National Gallery, London, NG189. Image record.
- **6:** Rembrandt, *Flora*, 1634, Hermitage Museum, Saint Petersburg, Image record.
- **7:** Albrecht Dürer, *Self-Portrait with Fur-Trimmed Robe*, 1500, Alte Pinakothek, Munich, inventory 537. Image record.
- **8:** Élisabeth Louise Vigée Le Brun, *Self-Portrait with Her Daughter, Julie*, 1786, Louvre, Paris, INV 3069. Image record.

Figure 2, query and destinations in stored order:

- **Query:** J. M. W. Turner, *The Fighting Temeraire*, exhibited 1839, National Gallery, London, NG524. Image record.
- **1:** Ivan Aivazovsky, *The Black Sea*, 1881. Image record.
- **2:** J. M. W. Turner, *Rain, Steam and Speed – The Great Western Railway*, 1844, National Gallery, London, NG538. Image record.
- **3:** Canaletto, *The Stonemason's Yard*, about 1725, National Gallery, London, NG127. Image record.
- **4:** John Constable, *The Hay Wain*, 1821, National Gallery, London, NG1207. Image record.
- **5:** John Martin, *The Last Man*, 1849, Walker Art Gallery, Liverpool, WAG2680. Image record.
- **6:** Caspar David Friedrich, *Monk by the Sea*, 1808–1810, Alte Nationalgalerie, Berlin. The release uses the pre-restoration reproduction. Image record.

- **7:** Claude Monet, *Impression, Sunrise*, 1872, Musée Marmottan Monet, Paris, inventory 4014. Image record.
- **8:** Théodore Géricault, *The Derby at Epsom*, 1821, Louvre, Paris, MI708. Image record.

Figure 8, query and destinations in stored order:

- **Query:** Edward Burne-Jones, *The Golden Stairs*, 1880, Tate Britain, London. Image record.
- **1:** Mary Cassatt, *In the Loge*, 1878, Museum of Fine Arts, Boston, 10.35. Image record.
- **2:** Gustav Klimt, *The Old Burgtheater in Vienna*, 1888–1889, Wien Museum, Vienna. Image record.

The checked-in `figure-data/retrieval-plates.json` preserves the release ID, ordered destination patch IDs, exact normalized crop bounds, selection rules, and raw 20-byte records. `figure-data/plates/run.json` records the source and derivative hashes. A normal figure rebuild uses those included derivatives; refreshing them requires the original local release and paintings.

Reproducing the new evaluation

`evaluation-config.yaml` fixes the query sample, seeds, fitting settings, and baseline descriptors. `evaluate_release.py` reads the installed release and its exact cached inputs; it writes only an isolated `evaluation/` directory beside this paper. It does not install its alternate clusterings. With the original Cabinet checkout and approved originals available:

```
.venv/bin/python docs/whitepaper/evaluate_release.py \
  --cabinet '/path/to/cabinet'
```

The run verifies the feature, index, label, and manifest hashes before analysis, and verifies each original image before extracting local descriptors. Every alternate clustering persists its centroids together with the installed PCA transform. `evaluation/run.json` records provenance and output hashes; `evaluation/results.json` includes all query IDs and per-query overlaps. Large feature and assignment arrays remain local, while the report and compact figure snapshot are included with the paper.

The validation record in `evaluation-validation/` checks three independent boundaries: the original seed reproduces all 686,116 installed assignments; preprocessed masks and coordinates agree with the cached index for all 308 paintings; and a scalar cosine reference agrees with the evaluation ranker on both unrestricted and cluster-restricted fixtures. PCA projection uses the original arithmetic order and block size, since algebraically equivalent floating-point operations need not reproduce a mini-batch clustering.

Rebuilding the product release

In the Cabinet project:

```
npm run echoes:check     # verify catalogue and image correspondence
npm run echoes:update    # extract, cluster, compile, validate, install
```

Assume every rebuild invalidates all labels. Old concept names are disabled on installation even when the new clustering looks similar, and all 200 groups must be reviewed against fresh crops before named findings return.

Appendix B: glossary

Patch. One 16x16 pixel cell of a preprocessed painting, and the atomic unit of both feature extraction and retrieval. 686,116 exist across the corpus.

Concept. One of 200 k-means clusters over patch features. Deliberately unnamed in the pipeline. Product words are assigned only afterwards; the current release records an assistant visual review.

Band yield. Clean, unflagged concepts recurring across 30 to 150 paintings, per concept produced. Used to select k, and treated as a search heuristic rather than as evidence (§9.5). Not comparable across corpus sizes.

Clean and flagged. A concept is flagged when it correlates with image position or image edge rather than with content, as frames, mounts, and scan margins do. Flagged concepts are demoted, not deleted.

Coherence. A reviewer's rating of a label group as *tight*, *mixed*, or *loose*, surfaced in the interface so it can decline to sound confident where the evidence was thin.

Durable. A concept retaining at least 50% of its members under every seed tried. 9 of 118 qualified on the historical 310-painting snapshot, and 18 of 131 in the new installed-corpus test. This criterion measures retention, not purity of the destination group.

Sentinel. The all-ones packed value, 1,048,575, meaning no match exists. It cannot collide with a real patch ID.

World A and World B. The metadata wall. World A, visual learning, never sees artist, title, date, or any human tag. World B, retrospective analysis, may join them only after concepts are frozen.

Appendix C: figures at a glance

```
CORPUS                      308 paintings, public domain, Wikimedia Commons
PATCHES                     686,116 (16x16, long side 896, aspect preserved)
BACKBONE                    DINOv3 ViT-B/16, frozen, layer -1, 768-dim
CLUSTERING                  MiniBatch k-means, k=200, PCA 128 (79.1% variance)

FEATURES (not shipped)      2.11 GB
ROUTES, uint32              21.96 MB
ROUTES, packed 20-bit       13.72 MB     <- 37.5% saving from packing
CONCEPTS                     0.69 MB
BINARY SUBTOTAL             14.41 MB     <- 146x vs raw float32 features
ALL FIVE RELEASE FILES      14.48 MB

QUERY                       20-byte read
REPORTED WARM LOOKUP        0.04 to 0.09 ms (local server, not HTTP)
COLD START                  ~45 ms
MODEL CALLS PER QUERY       0 (hosting costs excluded)

EXACT TOP-1 AGREEMENT       1.000  (100 queries; a test for implementation
MEAN TOP-8 OVERLAP          1.000   error, not an accuracy estimate)
COVERAGE (924 API QUERIES)   7.99 / 8 average, 0 empty walls
STORED-ANSWER CENSUS        685,141 records with 8; 975 with 2 (all records)

NEW EVALUATION              installed 308-painting release
STORED ROUTES RECHECKED      1,000 / 1,000 destination sets agree
SEED DURABILITY             18 / 131 clean concepts, all four seeds
HISTORICAL DURABILITY       9 / 118, separate 310-painting snapshot
WALL RETENTION              see Figure 10; 1,000 queries x four seeds
GLOBAL DINO AGREEMENT       54.35% exact patches; 74.04% paintings
LOCAL CONTROLS              RGB histogram, pixel cosine, HOG measured
                            Route agreement, not human relevance

NEXT QUESTIONS              relevance of changed walls; stable groups
HUMAN LABEL VALIDATION      not yet measured; labels assistant-authored
CLIP CONTROL / USER STUDY   not yet run
```

Prepared from project source and decision records, 16 September 2026; revised 17 September 2026. Figures use a checked-in evidence snapshot with source hashes. Some underlying experiments and images are local artifacts rather than committed data. Timings are attributed project reports, not new measurements. Negative and inconclusive results are retained. See figures.md for figure provenance and build instructions.