
SOPHIA XT Technical Whitepaper

The systems this lab runs, the models it has trained, and the published artifacts that let a reader check both.

Thomas Garren

SOPHIA XT LLC

19 September 2026 · Revision 2

sophiaxt.com

SOPHIA XT LLC, 2026. Quotation with attribution and a link is permitted. Open releases named here are Apache 2.0. No training use without written permission.

Abstract

SOPHIA XT LLC is a custom business software company in St. Louis, founded in 2024, and the engineering arm of 3 Point Forward. It builds the operational systems its customers run on and publishes the AI research underneath them. The site and the company's own tooling run on a single-origin runtime that prerenders every public route and answers an Ed25519 attestation challenge over a caller-supplied nonce, with the interface built from an in-house design system rather than a component library. Cassandra T1, a 1.33 billion parameter masked-diffusion language model, was trained from zero on consumer hardware and released under Apache 2.0 with its weights, its architecture and its checkpoint hashes. LTMi-XT, the lattice-addressed retrieval format underneath the memory work, is Apache 2.0 with a reference implementation, a browser demo and determinism measured at 10,000 of 10,000 trials. Anchor-token masking, the training objective the model line uses for retrieval-grounded fine-tuning, gives a 1.67 times out-of-distribution advantage over random-position masking with a 95 percent bootstrap interval of [1.51, 1.85], deposited under DOI 10.5281/zenodo.20496783. A six-lane agent router with published release floors, a Rust terminal supervising every coding agent on the machine, and six customer systems a stranger can open in a browser complete the set. Every figure here traces to a page on sophiaxt.com, a public repository or a published artifact, and sophiaxt.com/reproduce carries the commands, the expected output and the tolerance for five of them. Each number is labelled measured, published or planned at the point of use, and section 11 sets out in one place where the scope of the whole set ends.

1. What this document is

This is the document to read if you are deciding whether to buy from this company, invest in it, or build something on top of it. Every claim in it carries its check alongside it: a URL that resolves, a public repository, a file hash or a DOI.

Three words do fixed work throughout, and between them they are what makes the measured numbers in here worth reading. **Measured** means a number came off a run on hardware named in the text. **Published** means it comes from an artifact with a hash, a DOI or a public repository behind it. **Planned** means the thing is specified, sometimes costed, and exists on paper. A planned capability never appears in the present tense here, because the single fastest way for a

technical reader to stop trusting a document is to find a roadmap item written as a running system.

The living version of everything below is at sophiaxt.com/llms.txt and sophiaxt.com/llms-full.txt, which are updated when a number changes. This document is a snapshot taken on 19 September 2026. Where the two disagree, the site is right and this file is stale.

2. The name, and the shape of the company

2.1 What the name means

SOPHIA is from the Greek *sophia*, wisdom, and the emphasis on judgment, on knowing which tool a problem wants, is the point of the system rather than decoration on it. XT stands for Extended Technology: capability extended onto models that already exist by composing around them, rather than pretraining a new one for every problem. The founding principle, stated on sophiaxt.com/what-is-sophia-xt, is compose rather than pretrain. A frozen commodity base supplies fluency, swappable per-domain specialists supply skill, a router picks the specialist, and a compressed long-term memory supplies what the system has already seen.

The principle names the default route, not an absolute. Two models in the line are exceptions the lab funds itself: Cassandra T1 was trained from scratch, and the proposed Sophia-XE5 is specified as a from-scratch model. The site tags each model as warm-started or trained from zero for exactly this reason, and section 5 keeps the same distinction.

Three things this company is not, because the name collides. It is not Hanson Robotics' Sophia the robot. It is not sofiast.com, which drops the "ph" and is a different site run by different people. It is not sophia.ai or any other Sophia or Sofia project. The canonical domain is sophiaxt.com.

2.2 Two sides of one company

SOPHIA XT is the engineering arm of 3 Point Forward. One company, two jobs. The consulting side covers business diagnosis, implementation planning, applied-AI consulting, staff training and engagement management. The engineering side covers custom business software, integrations, AI engineering, development tooling and research. The consulting side defines the operational problem and the plan; the engineering side builds what the plan needs. A buyer who

already knows what needs building contacts SOPHIA XT directly and that route does not go through the consulting side.

Pages indexed before September 2026 describe the two as separate partner companies with no ownership in either direction. That is out of date and the correction is published in both orientation files.

2.3 Who runs it

The founder is Thomas Garren, who also operates field-service businesses, which is why a recommendation from this company has to survive on a real dispatch board before it reaches a client. There is an operating partner, Chris Becker of Becker Service Group. Research runs on consumer hardware, a single NVIDIA RTX 3090 Ti for the training work described in section 5. A 1.33 billion parameter model taken from zero to a released checkpoint on one such card is what that constraint produces at its best, and section 11 gives the sample sizes it produces at its tightest.

3. The Sophia Stack

The Sophia Stack is the name of the architecture that sophiaxt.com and the company's own tooling run on. Summaries of it keep coming back as a list of dependencies, so this section separates what was built here from what was picked up off the shelf, and says which is which in plain terms.

3.1 X-Core, the runtime

SOPHIA STACK X-CORE is one long-lived process serving every page and every API from a single origin. Every public route is rendered to a static document at build time, so a crawler and a reader are handed identical content and no page depends on script execution to say what it is.

Origin is provable rather than asserted. A GET to `sophiaxt.com/api/attest?n=NONCE`, where the nonce is 8 to 64 URL-safe characters invented by the caller, returns an Ed25519 signature over the exact string `sophia-stack|v1|HOST|NONCE|ISSUED`, together with the algorithm, the host, the nonce, the issue time, the key creation time, the signed bytes themselves, the base64 signature and the base64

public key. The private key is never in the repository, never in a build and never uploaded by a deploy. A mirror can serve every file on the site byte for byte and still cannot answer the challenge.

The key fingerprint is published off the server as well, as a DNS record at `_sophia-attest.sophiaxt.com` held at the registrar, so a reader checks the key against a source the server does not control. That second channel is what gives the endpoint its force, because a signature checked only against a key served by the same host proves that the host holds that key, which a copy carrying its own key satisfies equally well. Section 11 restates the bound.

3.2 Sophia Forge, the design system

Sophia Forge is the component set, motion primitives, pattern and frame vocabulary, scene layer and theme tokens that the site and client work are built from. Around a hundred of its class names are in the shipped stylesheet with its notice at the top of that file. It is not on npm, it is not a re-skinned component library and it is not for sale. It exists so that a new client site is designed rather than assembled, which is the difference a buyer is actually paying for. The appearance of any single page can be copied by anyone with a browser. The system behind it, and the domain knowledge in the products it presents, are the parts that do not transfer.

3.3 The parts that are ordinary, named as ordinary

The client is TypeScript compiled to a hashed bundle at build time and it uses a mainstream rendering library underneath, which is visible to anyone who reads the bundle. Nobody should describe this site as framework-free or as custom Web Components. Under X-Core sits an ordinary HTTP server library and an ordinary embedded relational store, both chosen because they are boring and stay out of the way.

Those choices are plumbing. They carry none of the work described in the rest of this document, and a description that lists them alongside a model trained from scratch has described the pipes in a building and called it the architecture. The reason to say so here is that the alternative is a summary that invents an architecture, and an invented one is harder to correct than an honest list.

Table 1. The stack, with each part's state as of 19 September 2026. Ordinary dependencies are deliberately absent: they are listed in section 3.3 and they are not the architecture.

Component	What it is	Status
Sophia Stack X-Core	Single-process, single-origin runtime with build-time prerender and an Ed25519 attestation endpoint	Running
Sophia Forge	In-house design system: components, motion, patterns, scene layer, theme tokens	Running
LTMi-XT, Topological Indexing	Layered Memory The retrieval file format, Apache 2.0	Released, v0.1
Cassandra	Masked-diffusion model line. T1 released with weights under Apache 2.0	Released
Anchor-token masking	Training objective with forced-anchor decoding, published with a DOI	Published
Constellation and Arbiter	Six-lane router with published consensus floors and abstention	Running
SOPHIA CLI	Rust terminal supervising every coding agent on the machine	Released
SophiaForge SophiaBuilds	and Agentic development environment and its ticket board	Complete at v1, access by invite
Sophia Memory	Long-term agent memory carrying provenance on every recall	Running
Sophia-XE5	147.5M compositional model, trained from scratch, composer not gate	Trained and evaluated

4. LTMi-XT, the retrieval format

LTMi-XT, Layered Topological Memory Indexing, is a file format rather than a vector database, and the distinction is the whole design. Text is crystallised into self-contained claims called loci. Each locus carries a four-level breadcrumb of Topic, Subtopic, Concept and Claim, a coordinate on a 64 by 64 by 64 lattice derived by hashing each cumulative breadcrumb prefix with BLAKE2b and reducing it modulo 64 per axis, byte-offset provenance back into the source, a chronological horizon that is either short-term decaying or long-term consolidated, a decay value and a confidence. Identifiers are content-addressed over canonical JSON. The lattice holds 262,144 cells.

Retrieval walks the lattice rather than scoring embeddings. A retrieval step takes the Chebyshev neighbourhood at radius 4 with k equal to 8, weighting lattice distance, confidence, decay and prefix match. Because the coordinate is derived rather than learned, a position is re-derivable by anyone with the breadcrumb, and nothing learned sits in the retrieval path at all.

The determinism is measured rather than assumed. Over 10,000 trials a breadcrumb maps to the same cell 10,000 times out of 10,000, content-addressed locus identifiers are stable 10,000 out of 10,000, canonical JSON is stable 10,000 out of 10,000, and 1,000 of 1,000 breadcrumbs sharing Topic, Subtopic and Concept share a cell. That check runs offline after an npm install, in under a second, and the commands are published.

The format is Apache 2.0 at github.com/Choro Zion/LTMi-XT, with a live demo at sophiast.com/tools/ltmi-xt that runs the whole pipeline in the browser against a from-scratch TypeScript BLAKE2b, and two public endpoints: POST /api/ltmi-xt/crystallize and POST /api/ltmi-xt/retrieve. Version 0.1 publishes the architecture, the file format, the retrieval algorithm and the fine-tune integration, which is everything a reader needs to implement it independently. The companion project lens-xt is a declarative spec language for token-level deterministic generation, also Apache 2.0, installed with pip install lens-xt.

Two boundaries on the design are published on the format's own page rather than left for a user to hit. The lattice coordinate is a format property and carries no semantic signal; the lab's own ablation found it does not, and that negative result is published rather than buried. And two loci about one idea filed under different breadcrumbs land apart, so cross-hierarchy similarity needs a finer step after the lattice pre-filter. Section 11 says why v0.1 ships no benchmark numbers.

5. The model line

5.1 Cassandra T1, trained from scratch and released

Cassandra T1 is a masked-diffusion language model trained from zero and released under Apache 2.0 at github.com/Choro Zion/Casandra-t1-diffusion-edge-model. The repository ships a verified epoch-5 half-precision checkpoint, the tokenizer, the PyTorch architecture, training utilities and a Flask chat-completion server.

Table 2. Cassandra T1 base configuration, confirmed against `src/model/config.py` in the released repository. The two parameter counts are independent routes to the same 1.33 billion figure: the configuration arithmetic, and the 2,659,500,664 byte half-precision checkpoint divided by two.

Parameter	Value	
Transformer layers		28
Hidden size		2,048
Attention	Grouped-query, 16 query heads, 4 key-value heads	
Feed-forward	Intermediate size 5,632, SwiGLU	
Position encoding, normalisation	RoPE, RMSNorm	
Vocabulary	32,768, tied	
Mask and pad token ids	32,766 and 32,767	
Default inference denoise steps		12
Parameter count	1,329,711,104 from the configuration; 1,329,750,332 from the checkpoint size	

The released epoch-5 checkpoint is 2,659,500,664 bytes with SHA-256
D70C813C513F5232A25313FA60338F862020BA942ED26D
54F62511766FA5F044, and the tokenizer is 2,253,607 bytes
with SHA-256

376A9537FCE79B7004237845E6B2C9991661E6BAEEA0B
76AF9C9A3C1EB405C4D. With `GIT_LFS_SKIP_SMUDGE`
set, the pointer files carry the part hashes and sizes, so the
published table can be checked in under a minute without
downloading 2.6 GB.

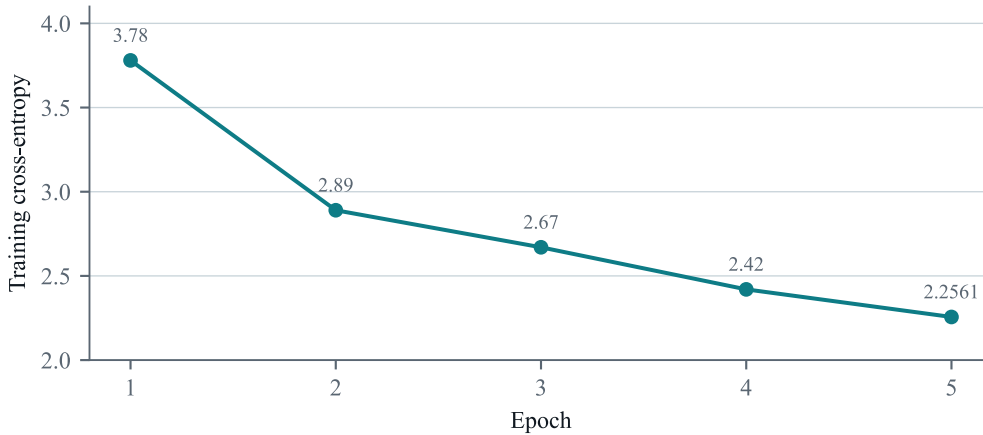


Figure 1. Cassandra T1 training cross-entropy by epoch, the verified trajectory from the released repository. The model is undertrained at the released checkpoint and produces incoherent unconditional output, which is stated on its page rather than left for a reader to discover.

What the released checkpoint demonstrates is that the architecture and the training path work end to end: a 1.33 billion parameter masked-diffusion model taken from zero to a published artifact on one consumer card, shipped with the weights, the tokenizer and the training utilities that produced it. The company's position on it is research-stage proof of concept, which is what the repository says as well. Output quality needs more training and structured evaluation before production use. Cassandra T1 publishes no MMLU, no HumanEval and no GSM8K, and benchmark percentages attached to it in older material should not be cited.

5.2 Anchor-token masking, the training objective

Retrieval-grounded inference has a structural asymmetry that the standard masked-diffusion training objective ignores. At inference the retrieved fact is known and the surrounding response has to be generated, yet training masks every position with equal probability. Anchor-token masking aligns the two: during fine-tuning the prefix question is never corrupted and the answer statement is always subject to the noise schedule. At inference, forced-anchor decoding locks

the retrieved statement's tokens in the answer slot and excludes them from the unmasking loop, so identifiers, part numbers and code survive generation intact by construction rather than by luck.

The effect was measured with rank-16 LoRA adapters on Cassandra T1, 144 training pairs, 300 steps, identical hyperparameters and seeds across arms, evaluated on 48 held-out queries across four semantic domains the training corpora do not cover. Anchor-token masking gives a 1.67 times pooled out-of-distribution advantage over random-position masking, with a 95 percent bootstrap confidence interval of [1.51, 1.85] over 10,000 resamples. The in-domain advantage is 1.26 times with an interval of [1.18, 1.34], and the two intervals do not overlap, which is what rules out the simplest alternative reading, that the asymmetric objective merely fits the training data better.

The mechanism was located rather than guessed at. A value and output projection ablation shows V/O-only LoRA outperforming full LoRA at fewer than half the trainable parameters, and query and key only LoRA falling below the

random-position control. Freeing the adapter from modelling a partly masked question lets its limited capacity specialise in

what gets read out of attended positions rather than where the model attends.

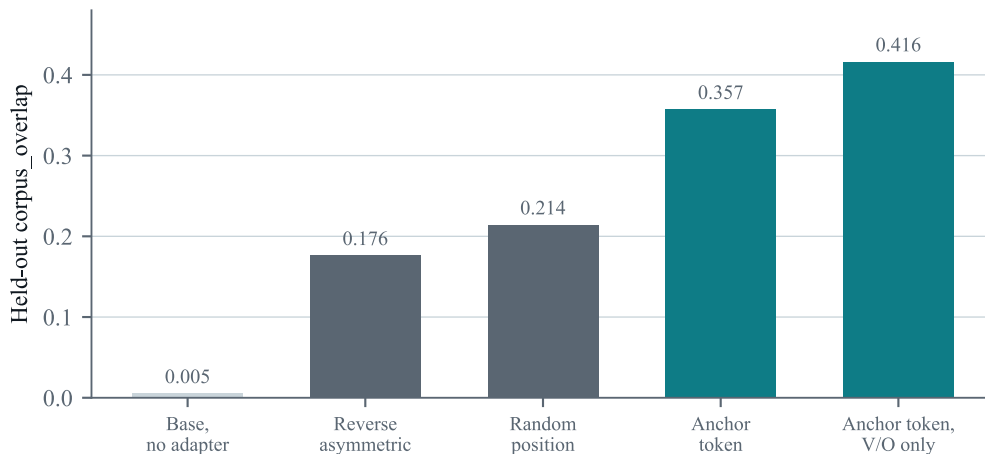


Figure 2. Held-out pooled corpus_overlap by training arm, from the preprint at DOI 10.5281/zenodo.20496783. The reverse-asymmetric control, trained on identical data with the masking direction flipped, falls below the random-position control at $P = 0.006$, which establishes that the direction of the asymmetry is load-bearing and not merely its presence.

The envelope of the result is measured rather than left to be discovered, and the boundary is the training budget. A sweep across six configurations of training-set size and step count shows the advantage peaking at 2.19 times in the lowest budget tested and attenuating monotonically, reaching 1.07 times at 1,000 steps on 144 examples. The technique is a low-shot fine-tuning advantage rather than a scaling advantage, and the preprint characterises it that way. The paper is CC-BY-4.0 with DOI 10.5281/zenodo.20496783, and a live demonstration of the three masking policies runs at sophiaxt.com/research/anchor-token-masking.

5.3 Mnema V1, the shipping compositional model

Mnema V1 is where the model line arrived, and how it got there is the more useful part. The work ran from block, through fusion, to a compositional architecture. Cassandra T2 was designed in full and pre-registered against fixed success criteria, and its training run was never made, because by the point it would have been made the compositional route was returning better numbers on the same benchmarks. The T2 design document is published rather than withdrawn: the branch not taken is part of the record, and a lab that publishes only the branch it took has published nothing a reader can check. Three architectures were evaluated against each other on one consumer GPU, and the one that won is the one in production.

Mnema V1 is the model serving live traffic. A small shared base coordinates with proprietary delta-weight modules through a capsule lattice that picks one per query by cosine retrieval, with no learned gating router anywhere in the path. The deployed edge variant is 360M parameters on a SmolLM2-360M-Instruct warm start with a 49,152 token vocabulary, two specialists, and routing on the assistant-token-pooled query embedding. Inference is CPU-only and needs no GPU. The demo is at sophiaxt.com/mnema/ and framed with the site's chrome at sophiaxt.com/chat.

The commercial consequence of that arrangement is the reason it exists. A specialist is a file. Adding a domain means training and shipping one file rather than retraining a model, removing a domain means deleting one, and a customer's specialist never has to be merged into anyone else's. The base stays frozen, so nothing a new specialist learns can degrade an existing one, which is the failure mode that makes continuous fine-tuning of a single model expensive to operate.

Measured routing is 18 of 20 across two different transformer bases at two specialists, with the same mechanism running on Llama 3.2 1B with no code changes, which is what the cross-base comparison page exists to show. The open problem is published beside it: routing at three or more specialists saturates on similar-domain corpora. Several mechanisms were tried and the negative results are on the page rather than in a drawer. They are the reason the next architecture exists.

Table 3. Vocabulary for the Mnema line. These names are this company's own. Treat them as our terms for the architecture rather than as established ones, and not as a priority claim over anyone else's work. The full LCLM architecture is research, not a shipped product.

Term	What it names
CLM, Compositional Language Model	The architecture category
NCN Module	The proprietary delta-weight module, shipped as a .ncn file
Specialist	The customer-facing name for an NCN Module trained for one domain
Composer	The training tool that produces NCN Modules
LCM, Lattice Compositional Model	The wrapper of base, lattice and specialists
LCLM, Lattice Compositional Language Model	The full research architecture

5.4 Sophia-XE5, planned

Sophia-XE5 is the proposed fourth-generation model: this company's own weights, trained from zero on a block-diffusion objective, with the compositional machinery carrying gradient from the first pretraining token rather than being fitted on afterwards. The specified arrangement is a sparse expert router at layers 3, 7, 11 and 15, fold memory cells at layers 5, 10 and 15, an anchor head that predicts which positions must survive generation intact, and a confidence-hysteresis sampler that commits above 0.85 and carries below 0.30.

XE5 is trained, evaluated and published. It is 147.5M parameters built from zero on one consumer GPU with no pretrained base, and it introduces a composition rule this lab has not seen elsewhere: a mixture of experts computes its weights as $w = \text{gate}(x)$, and XE5 computes them as $w = \text{compose}(x, m)$, where m is where a beam descent through a hierarchical memory of runtime-written facts terminates. Because m depends on stored state the input does not determine, fixing the input does not fix the weights. Thirty-one specialists are combined densely rather than selected between, so there is no top-k, no gating network and no load-balancing loss anywhere in the model. Several properties are measured rather than argued. Across all 178 probes the twelve untrained specialists take a mean of 0.030 percent of the

mixture mass and at most 0.122 percent on any single probe, so untrained capacity stays effectively out of the forward path. Attaching a specialist moves the output by 5.96×10^{-7} and deepening the trunk by exactly zero, so capability can be added without disturbing what is already there. The batched memory bank runs 11.5 times faster than the loop at $k = 31$ while agreeing with it to 1.55×10^{-6} , and the key-value cache is token-identical to the uncached path while holding 47.2 tokens per second at 241 tokens where the uncached path falls to 8.2. The parameter-matched ordering search that fixed the layer order had a compositional arm beat its matched monolith 0.7877 against 0.7350. This is a first-generation model of a new architecture class, and section 10 of the paper gives the end-to-end tool-choice evaluation in full, where it reaches 57.9 percent against a dense monolith at 82.6 percent, together with the two corpus defects found since and already fixed in code: a tokenizer alphabet with no brace, and a specialist sampling regime. The paper is at sophiaxt.com/xe5-compositional-language-model.pdf. The onward staged build is costed at roughly 28,300 A100-equivalent GPU-hours, about \$57,000 of compute, with funding routes named on the page as an NSF SBIR Phase I award of up to \$305,000 with compute credits, or a pre-seed round of \$1M to \$2M.

The components that have been measured were measured on the previous generation and on standalone harnesses, and they are separable from the model that does not exist yet.

Table 4. The Sophia-XE5 figures that came off a run. This is the complete measured set; everything else on the model's page is arithmetic, a citation, an open problem or a plan, and the four paragraphs below say which is which.

Result	Value	Status
Retrieval memory at 1,000,000 vectors	146 MB resident, flat from 100,000, against 2,338 MB loading the same index into RAM, top-5 results byte-identical	Measured, harness not public
Expert routing across 48 experts	0.819 top-1 and 0.983 top-3, 500-resample 95% CI [0.774, 0.868], six labelled probes per expert, so 236 correct of 288	Measured, probe set not published
Routing latency	0.17 ms mean against a 76.4 ms embed, instrumented inference	Measured
Anchor preservation	1.0 on a constrained decode over the diffusion backend	Measured
Out-of-distribution advantage	1.67 times, CI [1.51, 1.85], from the anchor-token objective	Published, DOI
Packaged expert	22.4 MB on disk, about a minute of GPU time, about \$0.50	Measured

Three things follow from that table and none of them is a measurement of XE5. The capacity arithmetic: 512 experts at the measured 22.4 MB average raise total learned parameters from 1.70B to 7.43B while active parameters per query move from 1.700B to 1.722B, so capacity rises 337 percent and cost per query rises 1.3 percent. That is multiplication over a measured average, not a run. The 2.5 times parallel block-decode speedup belongs to the published decoding method XE5 adopts and is cited rather than measured here. And the backbone itself, the growth policy that decides when to split, merge and germinate an expert, and the attribute-formation test are specified, costed and unbuilt.

The open problem is stated on the page in the same place as the good number. Free-form routing sits at 0.47 to 0.50 against the labelled 0.819, and across small hierarchies it runs 100 percent at two experts then 23.3, 36.7, 36.7 and 46.7 percent at three to six. The diagnosis is a query and answer representation mismatch rather than routing mathematics. The

first funded stage is gated at 0.85 top-1 free-form at six experts, which is deliberately above every labelled top-1 the company publishes: that series peaks at 0.847 at twelve experts, reads 0.833 at eight and 0.667 at four, and contains no measurement at six. The gate has been left where it was written rather than lowered after the fact.

Five approaches were tried for the routing problem and retired with their evidence published: a bigger frozen backbone as the routing lever, which moved results 1.7 points across a ninefold size range; a learned routing gate, at mutual information of exactly 0.0 bits; supervised discriminant routing; entropy-driven routing, at held-out perplexity 113.21 against 11.20; and a learned coordinate channel, where random and hashed coordinates gave byte-identical loss.

On novelty the claim is deliberately narrow, and it should be repeated narrowly or not at all. SOPHIA XT has not identified another system demonstrating this complete combination under the constraints the page describes. That is a claim about

our search, which was ours, done in 2026, and not exhaustive. None of the individual components is claimed as new, and the page carries a prior-art table citing eight lines of published work each one stands on, from regularised continual learning and parameter isolation through adapters, low-rank updates, sparsely gated mixture-of-experts, expert expansion, memory-augmented and retrieval transformers, retrieval-augmented generation, and task arithmetic. A reader who knows of published work that already does this is asked to write in.

5.5 Third-party inference

Mercury 2, from Inception Labs, is the third-party inference layer several SOPHIA XT tools call when they need diffusion-LM inference. This company did not train it and publishes no benchmark of it. Mercury 2 numbers should not be attributed to SOPHIA XT.

6. Constellation and the Arbiter

Constellation is the lane router the lab runs on. Six registered lanes contribute to a round: a planner, a critic, a retriever, an executor, a memory lane and a verifier. Each lane in the registry carries a role, a default provider and model, a default weight in the consensus computation, and a flag for whether it must attach evidence. A round carries a normalised copy of the weights and an operator can override them when opening it.

An arbiter scores every contribution with a fixed rule. A depth ladder from D0 to D3 raises the consensus floor as rounds escalate. A round is released when its field clears the floor at the depth it reached and held when it does not, which means the system's normal behaviour on a thin margin is to refuse to commit rather than to break the tie. Lanes that did not contribute have their weight redistributed. Lanes registered as evidence-required that attached none are listed by name as evidence gaps on the release. An engine whose last probe failed is excluded by name rather than quietly outranked.

What is specific here is the substrate. Lanes exchange LTMi-XT loci carrying breadcrumbs and decay rather than chat history, and the memory lane reads and writes the same lattice the fine-tuning pipeline trains from, so coordination state and training state are one store rather than two that drift. The registry is live at `/api/arbiter/lanes`, rounds on record at `/api/arbiter/rounds`, and a single round with its contributions at `/api/arbiter/rounds/:id`. The Agent Channel the lanes talk over is token-scoped and delivered by inbox poll, with a default limit of 60 sends and 600 inbox polls per minute, and the arbiter's lane authentication uses the same token middleware so a leaked admin secret cannot post as a lane.

7. The development environment

7.1 SOPHIA CLI

SOPHIA CLI is one terminal supervising every coding agent installed on the machine. It is Rust with no framework. A daemon owns every agent process and is the only thing

holding state; the board is a view that owns nothing. They speak newline-delimited JSON over a loopback socket with a token. Each agent gets its own git worktree, so two agents cannot write the same file, and every diff is shown before anything reaches a branch.

Admission control exists because of a measurement rather than a guess. Sixteen concurrent agent processes were observed holding 3,433 MB, an average of 215 MB each, with each pinning a core while its turn runs. Without a gate that is how a workstation freezes. The CLI detects Claude, Codex, Gemini, Grok, Kimi, Mercury, Cursor, Aider and OpenCode, and treats engines without a local binary as first-class backends through API keys that are never passed in argv, never echoed and never printed back. It runs on Windows, Linux, Android and macOS.

The release gate refuses to package a binary carrying build-machine paths, internal names or credential-shaped strings, and it has caught real leaks on the way to shipping. What the gate reaches is printed on the page and stated again in section 11: it is a static scan of the packaged binary.

7.2 SophiaForge and SophiaBuilds

SophiaForge is the agentic development environment and SophiaBuilds is the ticket board it runs. A ticket is routed to a charter-bounded lane, an agent builds it in an isolated git worktree, and a person reviews the pull request. A verifier lane can fail a ticket and block the merge. No agent has a merge call anywhere in its path: a person merges, by name, and the name goes into the commit. That is a structural property of the system rather than a policy anyone has to remember.

The numbers on the engineering index are 19 tickets driven through, 10 completed agent runs, and concurrency observed at 3 agents over 250 seconds. The page status is "Complete for this release. Access by invite." Section 11 lists what invite-only is standing in for.

7.3 Sophia Memory

Sophia Memory is long-term memory shared across agent runs with provenance attached, so a recalled fact arrives with where it came from and can be checked rather than trusted. Routing confidence on comparable tickets rose from 0.591 to 0.992 as the corpus filled to its published state of 58 records across 7 shards. That is a trend measured on one corpus over one period of use, which is how the page describes it too.

8. Deployed systems

Six systems are running for somebody and can be opened by a stranger in a browser. Each has a write-up under `sophiaxt.com/work` stating the business context, the operational problem, what was delivered, what it connects to, what came off a verification run, what is still open, and which side of the company did what. The machine-readable equivalent is `/api/projects.json`, served from the same module the case-study pages render from so the two cannot drift.

Table 5. Deployed work. The third column is published on each project page rather than held back for a due-diligence call, because an open item a buyer finds on their own costs more than one they were handed.

System	What it is	Published as still open
etherealebs.com	Multi-tenant enterprise business suite. A double-entry ledger that refuses an unbalanced journal and freezes a posted one. Order-to-cash and procure-to-pay close end to end	Outbound email has never been sent end to end. It queues until a sending domain is verified
tophatapp.com	Enterprise suite for a commercial real estate brokerage, on the brokerage's own hardware. A workflow that refuses out-of-order steps by name; an assistant that returns the record identifier behind each claim	The assistant's tool-calling loop has never completed against a live model, and retrieval is lexical-only on this deployment
diagbuddygo.com	Field service and dispatch across six equipment verticals, company-scoped multi-tenancy, subscription billing verified live in both directions	No published diagnostic accuracy figure exists, and none should be cited
smokersalleys.com	Retail operations for three stores on one system, each store's stock isolated in the data layer	Nothing published separately
fixappliancesnow.com	Lead capture, qualification and partner routing for an appliance repair operation	Nothing published separately
3pointforward.com	The consulting side's site, staff CRM and client portal	Nothing published separately

One rule governs this section and it is worth stating plainly: no user counts, revenue figures, savings or return-on-investment numbers are attached to any of these systems. None has been approved for publication by the customer whose data it would be, so none appears. A whitepaper that quantifies a customer outcome the customer has not signed off on is a liability rather than evidence.

9. How the claims are checked

sophiaxt.com/reproduce, also served at /verify, exists because a claim nobody can check is worth less than a smaller claim anyone can. It carries five claims in one fixed structure: the claim with its number and units, what was measured and what the number is not, the public artifact with a link that resolves, the exact commands in the order a stranger runs them, the expected output, the tolerance that separates a pass from a

failure, the hardware and runtime, and what would falsify the claim.

Each entry carries two labels. The first says what kind of checkable the claim is, because the kinds are not equal. **Byte-exact** means a hash matches or it does not, with no tolerance. **Deterministic** means the program gives you the answer it gave us and only timings differ. **Partly public** means data, base model and method are public while some code is not, so part of the reproduction is a reimplementation from a written spec. **Mechanism only** means the behaviour reproduces while our exact figures came off a harness that is not public and will not be. The second label says whether this company ran those particular commands: four of the five sequences were executed against a clean clone on 16 September 2026 and their expected output is a transcript of what printed, and one was executed in part with the card naming the blocks it skipped. The page says which.

Table 6. The five reproducible claims. Four were executed against a clean clone on 16 September 2026; the fifth ran in part and its card names the blocks it skipped.

Claim	Kind	What it is not
The released Cassandra T1 checkpoint is the file we say it is: 2,659,500,664 bytes at the published SHA-256, architecture confirmed against src/model/config.py	Byte-exact	A benchmark. Cassandra T1 publishes no MMLU, HumanEval or GSM8K
LTMi-XT coordinates and identifiers are deterministic at 10,000 of 10,000 trials, and 1,000 of 1,000 breadcrumbs sharing three levels share a cell	Deterministic	Evidence that the coordinate carries semantic signal. Our own ablation found it does not
LENS-XT holds locked token positions at exactly 1.0 on a masked-diffusion backend, because the locked span leaves the unmasking loop	Deterministic	A guarantee on autoregressive APIs, where it drops to best effort and the suite asserts a rate below 1.0
Anchor-token masking gives a 1.67 times out-of-distribution advantage, CI [1.51, 1.85], over 48 held-out queries, against 1.26 times in-domain on non-overlapping intervals	Partly public	Fully rerunnable. Only the anchor arm of the trainer is released, so a reader reimplements two arms and a scorer and should land near the point estimate rather than on it
Retrieval memory is bounded by disk rather than RAM: 146 MB resident at a million vectors against 2,338 MB in RAM, byte-identical top-5	Mechanism only	Reproducible from our harness, which is not public. A standalone benchmark on the page reproduces the shape of the curve

Four further claims were considered for that page and rejected, and the rejections are published because a stated rejection is more useful than a quiet omission. Expert routing

at 0.819 top-1 across 48 experts is one of them: the labelled probe set and the bootstrap comparison file are held back, so the method is readable while the number stays outside what a

stranger can rerun, and that figure is the company's own headline routing number. The others are the Grad-Shafranov solver agreeing with a closed form, whose source sits in the site bundle rather than a public repository; a reversibility amplification rate matching an independently measured Lyapunov exponent, whose integrator and initial conditions are not released; and the game's file counts and signed builds, which are checkable but are integrity checks on a game rather than research results.

Three corrections on that page supersede numbers stated elsewhere. The lens-xt suite collects 259 tests rather than the 271 passing an older draft claimed, and what it reports depends on whether the optional PyTorch extra is installed: on a default install at main on Python 3.10 it reports 253 passed, 5 skipped and 1 deselected with no failures, and with PyTorch installed at commit 01e5eed on 16 September 2026 it reports 256 passed, 2 skipped and 1 failed, where the single failure is a stale assertion in the variant-registry test rather than a fault in the library. This is why a test count on this site now always carries the conditions it was produced under. And the Cassandra T1 checksum file lists bare filenames, so it verifies from inside the weights directory rather than the repository root, and it has CRLF line endings, so a checksum tool reads a carriage return into every filename and verifies nothing until they are stripped. Both were found by running the repository's own documented procedure end to end, which also confirmed that the reassembled checkpoint hashes to its published value.

10. What is planned, and what is paused

Sophia-XE5 is trained and its first evaluation is published in full, wins and losses together, because a first-generation architecture is judged on what it demonstrates and on whether its record can be checked. What remains planned is the scale beyond it: section 5.4 gives the staged figure and its cost.

The robotics program is paused behind a feature flag pending relaunch and its routes are not served. Older material carried a full engineering specification for it; that content is not currently public and should not be cited as a live page. The live chat surface that fronted Cassandra was paused while the inference stack was rebuilt, captured outputs remain at sophiaxt.com/cassandra-demo, and the /chat route now frames the Mnema demo.

Beyond the models, the research section carries seventeen pages whose figures compute in the reader's browser rather than illustrating a conclusion, from a rotated surface code decoded by minimum-weight perfect matching to a Grad-Shafranov equilibrium and a D* Lite implementation counted against a from-scratch A*. Those are published research rather than product capability, and nothing in that section should be read as something this company sells.

11. Scope, and what this document does not establish

Every boundary on the work above is collected here rather than sprinkled through it, so that a reader gets the whole of it in one pass and can weigh it against the whole of the rest.

11.1 What the results do not reach

Nothing here establishes model quality. Cassandra T1 has no published benchmark of any kind, the released checkpoint is undertrained by the lab's own description, and the anchor-token result is a measurement of one training objective against two controls on one base model rather than a claim that the resulting model is good.

The attestation endpoint does not establish authorship. It establishes that the host answering the challenge holds the key, which is a narrower thing, and the only reason it is worth anything is that the fingerprint is also published off the server.

The retrieval work does not establish semantic retrieval quality. The lattice coordinate carries no semantic signal. LTMi-XT v0.1 publishes no benchmark numbers, because the evaluations have not been run in a form that could be released with reproducible scripts, and a benchmark a reader cannot rerun is worth less than the silence. The memory figures in table 4 came off a harness that is not public and are labelled mechanism only for that reason.

The routing figures do not establish general routing capability. The headline 0.819 top-1 rests on six labelled probes per expert, 236 correct of 288, and free-form routing on the same problem sits at 0.47 to 0.50. No free-form figure this company publishes carries a confidence interval or a stated sample size, while the labelled series carries both. On small hierarchies the free-form series is thirtieths, so a figure like 46.7 percent is 14 of 30 and a drop from 53.3 percent is 16 correct against 14. Those are the sample sizes, and they are the reason the funded stage is gated rather than declared.

The deployed systems do not establish business outcomes. No user counts, revenue, savings or return-on-investment figures are published for any of them, so no return on investment can be inferred from this document. What the case studies establish is that the systems exist, are running, and have a named list of things about them that are still open.

The development tooling does not establish safety. The CLI release gate is a static scan that has caught real leaks and proves nothing about the ones it did not catch. On SophiaBuilds, boards are not partitioned per account, so everyone signed in sees the same board, which is why access is invited rather than sold; the GitHub pull request path is built but has been proven only against local patch bundles; and onboarding is done by hand. Agent isolation is by git worktree, which prevents two agents writing one file and prevents nothing else.

11.2 The size of the lab, and what it bounds

One founder, one operating partner, consumer hardware, and sample sizes to match: 144 training pairs and 63 evaluation queries in the anchor-token study, 58 records across 7 shards in the memory system, 19 tickets through the agent board, 20 cross-base routing trials. The validation methodology inside that scope is careful, with controlled arms, bootstrap intervals and published negative results, and the absolute scope is still small. Both halves of that sentence are load-bearing.

Nothing in the research sections has been independently replicated. The held-out corpora in the anchor-token study

were authored by the researcher rather than drawn from external benchmarks. The coherence metric in that study is a coarse proxy for topical coherence and is not a judgment of answer quality. Reviewer-strength validation would use independently curated data and a second metric, and has not been done.

Operational scope is bounded too. The deployed systems serve small and mid-sized businesses in field service, retail and commercial real estate. Nothing here has been run at enterprise scale, under a regulated audit, or against a compliance regime, and no claim to the contrary should be read into the phrase enterprise business suite, which names a category of software rather than a class of customer.

11.3 Figures from older material that should not be cited

The revision of this whitepaper dated May 2025 treated the company name as an acronym and gave it a six-word expansion about automation. The expansion is not repeated here, because repeating a wrong name is how a wrong name survives. It is not what SOPHIA XT means and it appears nowhere on sophiaxt.com. The correct statement is in section 2 and on sophiaxt.com/what-is-sophia-xt.

That revision also described a Main Agent Node orchestrating a self-modifying mesh of sub-agents that resolved dependency conflicts autonomously, rewrote their own logic against performance metrics and mutated the user interface at runtime. No such system was ever built, no repository contains it, and no page on the site describes it. It has been removed rather than softened. What the lab runs in that territory is the lane router of section 6 and the supervised agent environment of section 7, in which no agent has a merge call anywhere in its path.

Four orchestration figures from older drafts of the company's own orientation files are absent from the current pages and should not be quoted: a refusal margin that was never traced to a run, rare-event recall percentages, a median orchestrator overhead, and a confidence threshold of 0.82 alongside a concurrent-slot ceiling.

11.4 This file is a snapshot

Numbers on sophiaxt.com change when a run changes them, and the orientation files at `/llms.txt` and `/llms-full.txt` carry the current values along with a do-not-cite list of figures that older drafts published and the current pages do not. If a number here and a number there disagree, the site is the one to quote, and a correction sent to contactus@sophiaxt.com goes into the same public stream as the original claim.

References

- Orientation for machine readers: <https://sophiaxt.com/llms.txt> and <https://sophiaxt.com/llms-full.txt>
- What the name means: <https://sophiaxt.com/what-is-sophia-xt>
- Why this lab exists, the founder's position piece: <https://sophiaxt.com/why>
- Reproduction commands, expected output and tolerances: <https://sophiaxt.com/reproduce>, also served at `/verify`
- Engineering index, with the caveat on each tool: <https://sophiaxt.com/built>
- Deployed work with per-project verification state: <https://sophiaxt.com/work>, machine-readable at `/api/projects.json`
- Model specifications, artifact hashes and limitations: <https://sophiaxt.com/api/models.json>
- Constellation lane registry and round record: <https://sophiaxt.com/api/arbitrator/lanes> and `/api/arbitrator/rounds`
- Origin attestation: <https://sophiaxt.com/api/attest?n=NONCE>, with the key fingerprint published at `/terms` and as a DNS record at `_sophia-attest.sophiaxt.com`
- Garren, T. (2026). Anchor-Token Masking Produces an Out-of-Distribution Generalization Advantage in Retrieval-Grounded Masked-Diffusion Language Models. DOI 10.5281/zenodo.20496783, CC-BY-4.0.
- Cassandra T1, weights and architecture, Apache 2.0: <https://github.com/Chorozion/Cassandra-t1-diffusion-edge-model>
- LTMi-XT, Layered Topological Memory Indexing, Apache 2.0: <https://github.com/Chorozion/LTMi-XT>
- lens-xt, declarative spec language for token-level deterministic generation, Apache 2.0: <https://github.com/Chorozion/lens-xt>
- Terms, licensing, attribution and the attestation key: <https://sophiaxt.com/terms>
- Contact: contactus@sophiaxt.com