

SOPHIA Custom: Training Methodology

Corpus composition, anchor-token-masked adapter training, what is held out, how a run is judged, and what a run writes down.

Thomas Garren

SOPHIA XT LLC

19 September 2026 · v1.1

sophiaxt.com

CC BY 4.0. Reference implementation is Apache 2.0.

Abstract

SOPHIA Custom turns a customer corpus into a domain-specialised adapter over a retrieval-grounded masked-diffusion base model. This document specifies the method with enough detail to follow it: how prose is crystallised into atomic loci, how training pairs are generated and where that generator is weak, the anchor-token-masking training objective and the adapter configuration it runs with, what is held out and why, the thresholds a run has to clear before it deploys, and the record a run leaves behind. The training objective is validated: anchor-token masking gives a 1.67x pooled out-of-distribution advantage over random-position masking with a 95 percent bootstrap interval of [1.51, 1.85], and adapters wrapping only the value and output projections beat full adapters at 2.98M trainable parameters against 7.0M. Both come from the study deposited under DOI 10.5281/zenodo.20496783. The document also carries the lab's nulls, because they determine how the method is run: more examples along an axis a corpus already covers bought nothing across two separate interventions, the lattice coordinate in the retrieval format was found empirically content-free as a conditioning signal, routing specialist adapters bought nothing over one adapter trained on everything in the one comparison that has been run, and the anchor-token advantage attenuates to nothing at long training budgets. What follows from those nulls is the method's central operating rule: coverage of question forms, not volume of examples, is what a corpus has to buy.

1. What this document is

SOPHIA Custom is a per-customer fine-tuning pipeline. A customer supplies a knowledge corpus, the pipeline

crystallises it into a structured retrieval format, trains a customer-specific adapter on the resulting pairs under a modified masking objective, validates it against a held-out slice of the same corpus, and deploys it behind an endpoint that decodes with retrieved facts locked in place.

This is the methodology specification rather than the service description. It is written for somebody who wants to reproduce the method, audit a run that was done for them, or decide whether the approach fits their corpus at all. Pricing, tiers and commercial terms are on the service page and deliberately not here, because a methodology document that carries a price list ages into a document nobody trusts on either subject.

Every number in this document is either measured, in which case it names the run or the artifact it came from, or it is a policy threshold chosen by this lab, in which case it says so. There are no projections. Where a figure would have needed one, the figure is absent.

2. The empirical basis

The training objective this pipeline uses was chosen from a controlled comparison rather than from intuition. Three masking strategies were trained on identical data with identical hyperparameters, an identical adapter rank and an identical 7.0M trainable parameter budget. Only the mask differed. Anchor-token masking never masks the question and applies the noise schedule to the answer statement at every step. Random-position masking applies it uniformly across both. The reverse-asymmetric arm masks only the question, and exists to test whether the direction of the asymmetry matters or only its presence.

Table 1. Pooled forced-anchor corpus overlap over 48 held-out queries across four domains none of the adapters was fine-tuned on. Measured, DOI 10.5281/zenodo.20496783.

Arm	Pooled held-out overlap	What it is
base, no fine-tuning	0.0045	the floor
reverse-asymmetric mask	0.176	the direction control
random-position mask	0.2137	the control
anchor-token mask	0.3571	the arm this pipeline runs

The pooled ratio of anchor to random is 1.67x with a 95 percent bootstrap interval of [1.51, 1.85] over 10,000 resamples and P(anchor greater than random) of 1.000. In

domain the same procedure gives 1.26x with an interval of [1.18, 1.34]. The two intervals do not overlap, so the advantage is larger out of distribution than in it, which rules

out the simplest alternative that an easier objective merely fits the training data better. The reverse arm sits below the random arm at P equals 0.006, so the direction of the asymmetry is load-bearing. The effect replicates at P equals 1.000 on astronomy, medical and legal held-out corpora and weakly at 0.994 on a cooking corpus, where the random-position arm is

anomalously high and a generic-vocabulary floor is the likely explanation.

A four-arm ablation then located where the effect lives, which is what determines the adapter configuration this pipeline uses.

Table 2. Two arms with the same 2.98M trainable parameters land a factor of three apart, because of where the parameters sit rather than how many there are. Measured, section 5.7 of the same study.

Adapter arm	Held-out overlap	Trainable	What it shows
random mask, all projections		0.214 7.0M	the control
anchor mask, all projections		0.337 7.0M	the reference arm
anchor mask, value and output only		0.416 2.98M	fewer parameters, better result
anchor mask, query and key only		0.136 2.98M	below the control

The account that survives all four probes in that study is not that the model learns to look at anchors more. Attention-pattern instrumentation found the shift in that direction to be about a tenth of the behavioural effect. What the objective does is free adapter capacity from the query and key projections, which no longer have to model a partially masked question, and spend it on the value and output projections, which control what gets read out of positions the model was already attending to. That is why this pipeline wraps only the value and output projections, and why doing so costs nothing in wall clock and saves more than half the adapter file.

3. What a corpus has to contain

The single most useful thing this lab knows about training small models is that what a model can generalise to is set by what its training data contains, and that volume along an axis

the corpus already covers buys almost nothing. That claim comes from a different model line in the same lab, a 17.5 million parameter tool-calling model rather than the masked-diffusion base used here, and it is stated as evidence about corpus construction rather than as a measurement of this pipeline. It is in this document because it determines how a customer corpus should be prepared, and preparing the corpus is the part of the method a customer controls.

3.1 Coverage decides what generalises

One model at one size was trained fifteen times. Each run had a single grammatical construction family removed from its supervised data and was evaluated on exactly the family it never saw, over 1,942 probes with procedural entities so that nothing was answerable from memory. Nothing about the model changed between runs. Accuracy on the withheld family ran from 100 percent down to 8.

Table 3. Selected rows from fifteen leave-one-out runs. The second seed changed the frames, the corpus and the probes, sharing no probe string with the first. Both extremes replicated and the middle did not, so anything between 40 and 90 should be read as roughly there rather than as ordered.

Withheld construction family	Accuracy on it	Second seed
passive, existential, possession	100 percent	100 percent
negated correction	100 percent	97 percent
need statement	92 percent	not rerun
interrogative	86 percent	not rerun
imperative	83 percent	not rerun
conditional	63 percent	not rerun
cleft, elliptical	53 percent	not rerun
reported speech	47 percent	65 percent
fronted adverbial	8 percent	8 percent

Four families cost nothing at all, because the intent sits in the verb and the entities and the construction carries no information the model needed. One family cost 92 points. A corpus that never states a request as a fronted adverbial produces a model that cannot read one, and no amount of additional data in the forms it already has will fix that.

3.2 Two nulls on volume

The same model was given more data in two different ways, and neither moved the held-out number. Growing the hand-written frame bank from 237 frames to 2,611, eleven times as

many drawn from families the bank already covered, left held-out tool choice unchanged on 257 probes. Adding 82 million tokens of human web text selected for construction coverage moved it from 48 percent to 49. The third intervention was different in kind: expanding the frame bank to generate across fifteen construction families, against the two the hand-written bank effectively covered, took held-out constructions answered from 5 of 24 to 10 of 24 with the model unchanged and the probe set fixed before the corpus was.

One further run moved both the corpus and the model at once, from 143 million to 310 million tokens and from 17.5 million

to 64.8 million parameters, and took held-out accuracy from 75 percent to 83 on 248 shared probes with twenty fixed and none broken. The comparison is paired and the sign is not in doubt, but because both changed together that run attributes nothing to either, and the experiment that separates them has not been run. It is reported here in the same terms.

3.3 What this means for a customer corpus

A corpus is prepared for coverage rather than for size. Before ingest, the useful question is not how many documents there are but how many distinct ways a user will ask for the same fact, and whether the corpus contains an example of each. Doubling the number of records that phrase requests the same way is the intervention that bought nothing twice. Adding a handful of records that phrase a request in a form nothing else in the corpus uses is the intervention that doubled coverage.

The practical consequence for this pipeline is in section 5. The default pair generator produces three question forms per locus, which is three forms no matter how large the corpus is, and that is the method's weakest point rather than its strongest.

4. Stage one: crystallisation

4.1 Input

Customer source material in PDF, plain text, Markdown, HTML, JSON records or a structured database export. The ingest ceiling is 100 MB of raw text equivalent per corpus.

4.2 Process

The crystalliser splits prose into self-contained atomic statements called loci. Each locus carries one fact, with

Q: Tell me about {concept} in the context of {topic}.
Q: What does the source say about {concept}?
Q: Regarding {topic}: {concept}?

compound claims split; has its pronouns resolved so that referents are explicit; reads on its own without the surrounding document; carries a byte-offset provenance link back to the source; and carries the crystalliser's own confidence in its faithfulness to that source. A bundle typically holds between 50 and 2,000 loci depending on input size.

4.3 Topology

Each locus receives a four-level breadcrumb of topic, subtopic, concept and claim. A three-dimensional coordinate on a 64 by 64 by 64 lattice is then derived by hashing breadcrumb prefixes with BLAKE2b, so loci sharing k breadcrumb levels share k coordinates. The lattice holds 262,144 cells. Identifiers are content-addressed over canonical JSON. Over 10,000 trials a breadcrumb maps to the same cell 10,000 times, identifiers are stable 10,000 of 10,000, and 1,000 of 1,000 breadcrumbs sharing topic, subtopic and concept share a cell.

That determinism is a format property and nothing more. It is what makes retrieval reproducible and identifiers stable. It is not evidence that the coordinate carries semantic signal, and a separate experiment, reported in section 12, found that it does not. The coordinate is an address.

5. Stage two: training pair generation

For each locus, three training pairs are generated by templated breadcrumb decomposition, giving three times the locus count. For a typical customer corpus that is between 150 and 6,000 pairs.

A: <locus statement>
A: <locus statement>
A: <locus statement>

Pairs are encoded with the question segment carrying a fixed system prompt ahead of the customer question, and the answer segment carrying the locus statement verbatim. The segment boundary is what the masking objective in the next stage keys on, so it is computed at encode time and stored with the pair rather than inferred later.

This generator is the weakest component of the method and section 3 is why. Three templates are three question forms. Two of them are declarative requests and one is a bare topic reference, and none of them is a conditional, a fronted adverbial, an elliptical follow-up or an oblique request, all of which are things users type. The leave-one-out result says a form the corpus never contains is a form the model answers badly, and multiplying the corpus by ten does not add a form. Where a customer can supply real questions from their own support logs, search logs or ticket history, those belong in the training set alongside the generated pairs, and a hundred real questions in varied forms is worth more than another thousand generated in the same three. This is an open weakness rather than a solved problem: the pipeline does not yet measure the construction coverage of a customer corpus, and building that

measurement is the next substantive change to this methodology.

6. Stage three: adapter training

6.1 The objective

Anchor-token masking restricts the diffusion masking schedule to answer-statement positions. Question positions stay visible at every training step. Padding is never masked. Loss is computed only over masked answer positions. Formally, for position i , the mask is 0 when i is in the question segment, drawn from a Bernoulli with the step's mask ratio when i is in the answer segment, and 0 when i is padding. The mask ratio is sampled uniformly from [0.5, 0.9] at each training step.

The inference-time access pattern is what this aligns with. Under forced-anchor decoding, the retrieved fact is known and locked and the surrounding response is generated, so training with the question fixed and the answer noised puts the model in the position it will actually be in.

6.2 Adapter configuration

Table 4. The adapter configuration, which is the study's winning arm rather than a default.

Parameter	Value	
Wrapped projections	value and output only	
Frozen projections	query and key, at base weights	
Rank		16
Alpha		32
Trainable parameters	2.98M, against 7.0M for a full adapter	
Initialisation	A by Kaiming uniform, B at zero	
Base model	frozen throughout	

6.3 Hyperparameters

Table 5. Identical to the configuration the masking study validated, so that a customer run and the published comparison are the same procedure.

Hyperparameter	Value	
Optimiser	AdamW	
Learning rate	1e-4 with cosine decay to 1e-5	
Betas	0.9 and 0.95	
Weight decay		0.01
Microbatch size		2
Gradient accumulation	1 to 2, effective batch 2 to 4	
Sequence length		192
Training steps	100 to 300, default 200	
Gradient norm clip		1.0
Mask ratio range	uniform over [0.5, 0.9]	
Random seed	configurable, 42 by default	

6.4 Training budget

Short budgets are not a cost decision. The sample-efficiency sweep in the validating study found the anchor advantage largest at the lowest budgets, between 1.97x and 2.19x at 100

steps, attenuating monotonically with step count and collapsing to 1.07x at 1,000 steps on 144 examples. The technique is a low-shot fine-tuning advantage rather than a scaling advantage, and training past the peak converts it into an ordinary fine-tune that has also overfitted.

Table 6. Recommended budgets. The first two rows sit inside the measured regime. The last two are extrapolations from the shape of the sweep and are governed by the held-out check rather than by the number in the table.

Corpus size	Steps	Basis
under 50 loci		100 at or near the sweep's peak advantage
50 to 150 loci	200 to 300	the regime the study measured directly
150 to 500 loci	300 to 500	extrapolated; held-out validation decides
over 500 loci	500 upward, with early stopping	outside the measured regime

For every corpus size the operating rule is the same: prefer the shorter budget, watch the held-out metric, and stop when it plateaus or turns down. A run that improves on training loss while the held-out metric falls is finished, whatever the step budget said.

6.5 Compute

The validated configuration trained in 75 to 100 seconds per adapter at 300 steps on a single consumer 24 GB GPU, and the wall clock is identical to a full-projection adapter on the same data; only the masking schedule and the wrapped projection set change. Full reproduction of the study, training and evaluation together, is about 60 minutes on the same

hardware. Figures for other hardware are not published here because none were measured.

7. What is held out

A held-out set that overlaps the training set measures memorisation and reports it as generalisation. Three separate held-out designs are used in this method, and they answer different questions.

The customer-facing hold-out is a slice of the customer's own loci, 10 percent by default with a floor of 5 loci, removed before pair generation so that neither the statement nor any question derived from it reaches training. It answers whether the adapter acquired the customer's topical bias or memorised the pairs it saw. It is the weakest of the three designs, because

held-out material from the same corpus shares vocabulary and structure with the training material.

The methodology hold-out, used when the pipeline itself is being evaluated rather than a customer adapter, is a set of corpora authored to be semantically distant from the training corpora. The four used in the validating study overlap the training vocabulary by 15.6, 11.5, 10.5 and 13.9 percent. This is the design that produced the 1.67x figure, and the gap between it and the 1.26x in-domain figure is the reason out-of-distribution performance is the number this method is tuned on.

The construction hold-out is the strictest and is used in the lab's corpus-composition work: probes verified disjoint from every training frame at the content-bigram level, with procedural entities so that nothing is answerable from

memory, and the probe set fixed before the corpus is changed. Fixing the probe set first is the part that matters. A probe set assembled after a corpus change measures the change's author.

One honest weakness runs through all three. The held-out corpora in the validating study were authored by the researcher rather than drawn from an external benchmark. They are factually checkable and semantically distant, and they are still not independent data. Reviewer-strength validation would use independently curated benchmarks, and that has not been done.

8. How a run is judged

8.1 Deployment thresholds

Before an adapter deploys it has to clear three gates. These are policy thresholds chosen by this lab, not measurements.

Table 7. Anchor preservation is structural on a masked-diffusion backend, since locked positions are excluded from the unmasking loop, so a value below 1.0 indicates a fault in the decode path rather than a weak adapter.

Gate	Threshold	What failing it means
Anchor preservation		1.0 the decode path is broken, not the training
Held-out corpus overlap	above 0.20	training failed to acquire the domain
Held-out keyword hit	above 50 percent	expected terms are absent from output

A run that misses a threshold is retried with adjusted settings, typically a different mask ratio range or a different step budget. If no configuration clears the gates the customer is told that, rather than being handed the least bad adapter.

8.2 The comparison arm

Every customer run also trains a full-projection, random-position-masked adapter on the same data and reports both sets of held-out metrics. The point is not marketing. It is that a customer whose corpus does not benefit from this objective should be able to see that in their own numbers, and the low-shot regime the technique lives in means some corpora will not benefit.

8.3 The statistical procedure

Comparisons between arms are reported as a point estimate with a bootstrap confidence interval rather than as two bare numbers. The published study draws 10,000 resamples from the per-query metric values and reports the percentile-method 95 percent interval on the ratio of means, together with the fraction of resamples in which one arm exceeds the other. An improvement is claimed only when the intervals do not overlap. A customer corpus small enough that the interval spans 1.0 gets that reported as an inconclusive result, which is the common case at the floor of 5 held-out loci.

8.4 The control that is easy to forget

If an evaluation changes the prompt format as well as its content, a control arm has to carry the format change on its own or the comparison charges the format to the content. This is not hypothetical here. In the retrieval comparison in section 11, putting an irrelevant record into the retrieval prompt format scored worse than the plain base model: the format alone costs 0.0257 in mean negative log likelihood, and relevance is worth 0.1147 on top of that. Without the control

arm, the retrieval result reads as 0.089 of benefit when 0.0257 of it is the format. The same rule applies to any comparison in this method that changes how a prompt is built.

9. What a run writes down

A result file that records only the headline means cannot be audited later, and this lab has published one that made exactly that mistake. The retrieval comparison in section 11 wrote out five arm means and six per-category means and nothing else. The base checkpoint, the seed, the adapter rank and the held-out record count were defaults of the script rather than values the file carried, so reading them means reading the script at the commit it ran, and the site says so rather than quietly letting the numbers stand alone. The run record below is what that experience produced.

- Corpus identity: the bundle hash, the locus count, and the crystalliser version that produced it.
- Hold-out identity: which loci were held out, by identifier, and the rule that selected them.
- Base model identity: the checkpoint file hash, not its name. Names get reused.
- Adapter configuration: rank, alpha, the exact set of wrapped projections, and the trainable parameter count as counted rather than as expected.
- Objective: the masking strategy, the mask ratio range, and the segment boundary rule used to split question from answer.
- Optimisation: learning rate schedule, betas, weight decay, effective batch, sequence length, step count, gradient clip.
- Seeds: the training seed and the generation seed, separately, because they move different things.

- Per-query metric values for every held-out query, not only the means, because an interval cannot be recomputed from a mean.
- The comparison arm's configuration and its per-query values, under the same rules.
- Environment: wall clock, peak memory, library versions, and the hardware the run happened on.

The test for whether the record is sufficient is simple. Somebody holding only the run record and the public artifacts should be able to rebuild the interval, not just read it. If they cannot, the record is short of something.

10. Stage four: deployment and inference

The trained adapter is loaded onto an inference endpoint issued per customer with key authentication. The interface carries four operations: a grounded query, the same query streamed, a retrieval-only call that returns the top loci without generating, and direct inspection of a single locus by identifier. Retrieval-only and locus inspection exist because a customer who cannot see what was retrieved cannot debug a wrong answer.

Inference uses forced-anchor decoding by default. When a query matches a high-confidence retrieval target, the retrieved

locus tokens are locked at fixed positions of the response and the unmasking loop skips them at every denoising step. Preservation is structural rather than statistical: a position the loop never visits cannot be altered by temperature, by sampling or by a hallucination mode. This is what makes the method suitable for content where an identifier has to survive exactly, such as part numbers, citations, dosages and dates. On autoregressive backends the same specification degrades to best effort, and the published test suite asserts a preservation rate below 1.0 there rather than claiming the guarantee.

11. Training the weights against retrieving into the prompt

A customer choosing this service is choosing to put knowledge in weights rather than in a prompt, and the honest answer to which is better is that it depends on what the knowledge is. One comparison has been run here and it is worth reporting in full, including the part that went the other way.

A 22,211-record corpus was cut semantically into six categories. One base model was held frozen and five arms were scored as mean negative log likelihood on the answer tokens of held-out records, 25 per category by the script's default, so 150 records in total. Lower is better.

Table 8. One run, one seed, six categories, no confidence intervals anywhere in it. Nothing here is a significance claim and intervals are the next thing to fix.

Arm	Mean answer NLL	What it is
base, irrelevant context in the prompt	1.1440	the format control
base alone	1.1183	the floor
base, retrieved answer in the prompt	1.0292	retrieval
base, routed specialist adapters	0.9258	one adapter per category
base, one adapter over everything	0.9238	the generalist

Both trained arms beat retrieval overall. The finding that matters more is where they did not. Of the six categories, five are procedures: SQL against a named table, Python and JavaScript functions, programming across languages, arithmetic word problems, and symbolic mathematics worked in prose. Adapters beat retrieval on all five, by between 0.118 and 0.476. The sixth is open-domain factual questions, and retrieval beats the routed adapters on it by 0.566.

Retrieval wins where the answer is a fact somebody already wrote down and loses where the answer has to be worked out. That category is also the only one where retrieval earns anything at all: drop it and the retrieval arm sits at 0.9218 against 0.8864 for the untouched base, so on the five procedural categories pasting a neighbouring record into the prompt was worse than leaving the model alone. It is also the loosest category by the silhouette score that selected the cut, the largest, and the one every arm finds hardest, with the base at 2.2777 against 0.73 to 1.03 everywhere else. A category that diffuse is exactly where a nearest-neighbour lookup has something to add and a specialist has the least shared structure to learn, so the loss is legible rather than mysterious.

The guidance that follows for a customer corpus is direct. A corpus of procedures, policies and worked methods is what

this method is for. A corpus that is mostly a collection of unrelated facts is a retrieval problem first, and the honest recommendation is retrieval with forced-anchor decoding over the base model rather than a fine-tune. Most real corpora are both, which is why the deployed system retrieves and decodes against a trained adapter rather than choosing.

12. Published nulls

These are the results that did not work. They are published because they are the reason the positive results above are worth reading, and because each one changed how this method is run.

12.1 The lattice coordinate carries no conditioning signal

The retrieval format underneath this pipeline puts every locus at a lattice coordinate derived by hashing its breadcrumb. A separate exploratory experiment asked whether that coordinate could additionally serve as a conditioning signal fed into attention. Three coordinate schemes were compared under identical architecture, data, warm start, optimiser and a 500-step schedule: the BLAKE2b hash, a principal-component projection of frozen-encoder embeddings, and uniform-random assignment. All three arms landed at the

same training loss of 0.2834, and unforced inference came out byte-identical between the random and hash arms with a zero-

width bootstrap interval.

Table 9. Paired-by-query bootstrap, 2,000 resamples, 36 held-out corpus and query pairs across three domains. The principal-component scheme reaches 96.7 percent one-nearest-neighbour topic accuracy in coordinate space against 21.7 percent for the hash, and that geometric advantage transfers to no downstream metric.

Comparison	Metric	Point	95 percent interval	Significant
PCA against BLAKE2b	forced overlap	-0.002	[-0.034, +0.029]	no
PCA against BLAKE2b	unforced overlap	exactly 0	[0.000, 0.000]	no
random against BLAKE2b	forced overlap	-0.016	[-0.054, +0.022]	no
random against BLAKE2b	unforced overlap	exactly 0	[0.000, 0.000]	no

What the experiment settled is narrow. The lattice is an address rather than a semantic geometry, and this lab stopped saying otherwise. It says nothing for or against the retrieval format, which was not what the experiment tested and which has not been benchmarked against alternative retrieval approaches. That gap is open. The same findings document publishes the adversarial review of the null, including the criticism that 36 pairs is too few against a noise floor of about 0.02 and that 200 would be needed for the interval the lab wants; the following round moved to 172. It also records prior art that was missed and names it.

12.2 Routing bought nothing over one adapter

In the comparison in section 11, the routed set of six specialist adapters finished at 0.9258 against 0.9238 for a single adapter trained across the whole corpus. Routing is ahead on four categories and behind on two, and the two it loses are the wider gaps, so the single adapter finishes 0.002 in front. On that corpus, on that run, specialisation and routing bought nothing over training one adapter on everything. This is reported because the architecture that motivated the experiment is more interesting than its result, and reporting only the interesting half is how a lab stops being believable.

12.3 The advantage attenuates with budget

The anchor-token advantage is not free at any scale. It peaks between 1.97x and 2.19x at 100 training steps and collapses to 1.07x at 1,000 steps on the same 144 examples. The 1.67x headline is a mid-regime measurement, not a maximum and not a floor. A customer with a very large corpus, past roughly 10,000 loci, should expect the benefit relative to standard fine-tuning to be smaller than the headline, and the method does not claim otherwise.

12.4 Volume along a covered axis

Restated here because it belongs with the other nulls: eleven times the training frames within families already covered moved held-out tool choice not at all, and 82 million tokens of construction-selected human text moved it one point. Two separate, expensive interventions that bought nothing. The corpus preparation guidance in section 3 is what those two nulls paid for.

13. Privacy and data handling

- Customer corpora are processed in isolation. No corpus is used as training data for another customer's adapter or for a

shared model.

- Crystallisation may call third-party language-model APIs for the topologiser step, in which case customer text passes through those APIs under their terms. A customer who cannot accept that takes the on-premise route below.
- Trained adapters are customer-specific and are deleted on request.
- Provenance is preserved: every locus carries a byte offset back into the source document, so an answer can be traced to the sentence that produced it.
- The full pipeline can run inside customer infrastructure where the corpus cannot leave it.

14. Limitations and scope

The technique is a low-shot fine-tuning advantage rather than a fundamental scaling advantage. Long training runs erode it, and the erosion is monotonic in step count over the range that was swept. The regime where it is worth using is short fine-tunes on small to medium corpora.

The validating evidence is one base model at 1.3 billion parameters, undertrained at cross-entropy 2.2561, at one adapter rank, on 144 training pairs and 63 evaluation queries across five corpora. Behaviour at larger scale, at other ranks, and on fully-trained base models is uncharacterised. Nobody outside this lab has replicated it.

Corpus overlap, the metric the gates are set on, is a coarse proxy. It measures whether generated text draws on the vocabulary of the topical domain. It does not reward grammatical structure, it does not penalise correct topical words in an incoherent order, and it is not a factuality measure. A model-as-judge replication would strengthen the interpretation and has not been run.

The method is tested only on masked-diffusion base models. The autoregressive analogue is prefix-language-model training, which is a different procedure with a different implementation, and none of the numbers here transfer to it.

The corpus-composition evidence in section 3 comes from a different model line at 17.5 and 64.8 million parameters on a tool-calling task, not from this pipeline. It is the reason the corpus guidance is what it is, and it is not a measurement of a customer adapter. The specific claim that construction coverage governs generalisation in this masked-diffusion pipeline has not been tested directly, and the pipeline does not yet measure construction coverage of a customer corpus at all.

The retrieval comparison in section 11 is one run, one seed, six categories and no intervals. It is enough to say that both trained arms beat retrieval overall on that corpus and that the one factual category went the other way. It is not enough to generalise to a corpus that has not been cut the same way, and whether the factual-category result survives intervals, a second seed and more than six categories is unknown.

Everything above was produced by a one-person research operation on one consumer GPU, and every sample is small. The controls are real, the intervals that exist are computed rather than asserted, and the nulls are published in the same place as the positives. Within that scope the evidence is good. Outside it, the right word for these results is suggestive.

15. Versioning

This document specifies methodology v1.1. Customer deployments record the methodology version they were trained under, so that an adapter trained a year ago can be audited against the procedure that actually produced it rather than against the current one.

Changes from v1.0, published 9 May 2026. The corpus-composition section, the held-out designs, the run record and the published nulls are new. The retrieval comparison is new. The pricing table is removed and lives on the service page. Hardware figures for configurations nobody measured are removed, as is an integration-time estimate that was not measured either. Third-party providers are described by role

rather than by name. The trainable parameter count is corrected from approximately 3M to the counted 2.98M, and the extrapolated rows of the training-budget table are labelled as extrapolations.

References

- Garren, T. (2026). Anchor-Token Masking Produces an Out-of-Distribution Generalisation Advantage in Retrieval-Grounded Masked-Diffusion Language Models. SOPHIA XT LLC. DOI 10.5281/zenodo.20496783.
- Garren, T. (2026). LTMi-XT v0.1: A Retrieval Format with Hash-Derived Topological Indexing. SOPHIA XT LLC specification.
- LTMi-XT empirical findings, 12 May 2026. Coordinate-scheme ablation with its adversarial review, published in the LTMi-XT repository.
- Data composition, not parameter count. SOPHIA XT LLC position paper, 18 September 2026, at sophiaxt.com/research/data-composition. Carries the leave-one-out construction ablation, the two volume nulls, the confounded scaling run and the retrieval comparison, each linked to the feed card, result file or script that produced it.
- Reproduction record, at sophiaxt.com/reproduce. Five claims with the artifact, the commands, the expected output, the tolerance and what would falsify each.
- Hu, E., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022.
- Raffel, C., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR 21(140).