LispM LLM: Scale-Invariant Open-Vocabulary Command Routing and Poison-Resistant Continual Learning for Symbolic Agent Harnesses

Ryan Kung
LispM Project (`lispm.ai`)
huggingface.co/ElderRyan/lispm-llm

November 2026

Abstract

Interactive symbolic environments such as GNU Emacs and Lisp Machine (*LispM*) agent harnesses expose hundreds of deterministic editor and application commands, yet routing natural-language instructions to those commands faces a sharp safety–flexibility dilemma. Frontier autoregressive language models synthesize arbitrary Lisp code at hundreds of milliseconds per call while introducing prompt-injection and `eval`-execution hazards; conversely, tiny closed-set intent classifiers guarantee sub-millisecond bounded dispatch without `eval`, but freeze their output layer to a fixed action vocabulary and remain static when users phrase commands idiosyncratically or load new packages.

We present **LispM LLM**, a local Rust + LibTorch command-routing architecture that resolves both limitations within **264,322 parameters** (≈ 1.0 ms per request on two CPU threads, zero GPU required). First, we formalize an *Open-Vocabulary Dual-Tower Bi-Encoder* over a zero-`eval` typed `ActionCatalog`, proving that naive cosine bi-encoders suffer from *reject-prototype domination* on unseen open actions, and introducing a *Scale-Invariant Semantic–Lexical Bilinear Decomposition* in $\mathbb{R}^{2d}$ that simultaneously admits unseen actions zero-shot, rejects unmatched out-of-domain nouns, and blocks inverted/destructive verbs on matched nouns via a single cached matrix multiplication. Second, we introduce a *Dual-Speed Poison-Resistant Continual Learning* loop that forbids raw self-training: verified external supervision from behavioral user corrections and higher-tier teacher agents updates a sub-millisecond hippocampal episodic store (`episodic.json`) through a monotonic *join-semilattice quarantine ledger* ($\mathsf{Consistent} \sqcup \mathsf{Obs} \to \mathsf{Conflicted}$), enforces automatic PII literal scrubbing and held-out benchmark leakage blocking, and consolidates durable weights during editor idle periods behind non-regression gates and an online circuit breaker. Across frozen regression suites (2,542 supported and 243 unsupported cases), an independent 277-case blind evaluation, and 30 real isolated Emacs 31.1 state-execution tasks, the open bi-encoder achieves **30/30 (100%)** real-task completion, **66/71 (93.0%)** blind two-action composition accuracy, reduces supported wording errors from 6 to **2**, and routes newly harvested `lispm-kit` and Emacs commands zero-shot under major-mode masking.

1 Introduction

Symbolic computing environments—exemplified by GNU Emacs and the modern LispM application suite (`lispm-finder`, `lispm-browser`, `lispm-stickies`, `lispm-rotom`, `lispm-mail`, `lispm-1password`, `lispm-security`, `lispm-shrdlu`)—already possess clean, composable functional primitives. When a user types *"open file `"notes.el"` then split right"* or *"copy the totp verification code"*, the user does not need a 70-billion-parameter model to invent new Emacs Lisp source code from scratch; they need a fast, auditable router that maps bounded English clauses onto verified public commands while refusing out-of-scope or ambiguous requests.

Existing approaches occupy two unsatisfactory extremes:

1. **Autoregressive Code Synthesis:** Prompting a cloud or local general-purpose LLM to emit S-expressions provides open-ended coverage, but incurs 200–2,000 ms latency, consumes gigabytes of memory or external API quota, and creates an inherent code-execution boundary hazard whenever generated Lisp is passed to `read` and `eval`.
2. **Closed-Set Intent Classifiers:** A compact encoder with a K-way linear classification head (such as the 184,625-parameter, 48-action baseline of LispM LLM) runs in ~ 1 ms on CPU and dispatches strictly through a static lookup table without `eval`. However, adding a 49th command requires changing tensor dimensions and retraining from scratch, and idiosyncratic user phrasing fails repeatedly until manual dataset curation occurs.

Upgrading a bounded command router from a static closed classifier to an *open-vocabulary, self-improving system* exposes four subtle failure modes:

- **Failure Mode 1 (Zero-Shot vs. Rejection Collapse):** During training on a base corpus of M_0 actions plus explicit `reject` examples (e.g., *"open spotify"*, *"copy line"*), negative prototypes learn to fire strongly on out-of-vocabulary ($\langle$`unk`$\rangle$) tokens and verbs absent from the original M_0 set. When a new action a_{M_0+1} is added at runtime without retraining, a naive bi-encoder's learned rejection prototypes overpower subword similarity, falsely refusing the newly registered command.
- **Failure Mode 2 (Confirmation-Bias Poisoning):** If a continual-learning loop treats high-confidence model predictions as pseudo-labels, early misclassifications reinforce themselves into permanent weight corruption.
- **Failure Mode 3 (Ambiguous Slang Drift):** Users occasionally apply the same shorthand phrase to different commands across contexts; naive append-only datasets oscillate or corrupt shared representations.
- **Failure Mode 4 (Literal PII Leakage):** Interactive commands routinely contain private file paths, URLs, credentials, or search strings that must never persist in training ledgers or model weights.

Contributions. This paper describes the complete mathematical and systems design of LispM LLM v0.3, shipped as open source and open weights under AGPL-3.0:

1. **Zero-`Eval` Open Action Basis (§2):** A typed `ActionCatalog` unifying the 48 audited core operations with dynamically harvested `lispm-kit-define-commands` mode tables and interactive Emacs commands under public-symbol, arity, string-domain, and major-mode invariants.
2. **Scale-Invariant Dual-Tower Bi-Encoder (§3):** A 264,322-parameter architecture combining a shared word + character-n-gram Transformer with a scale-invariant semantic–lexical bilinear embedding in $\mathbb{R}^{2d}$ and a 16-prototype negative-intent bank, accompanied by an analytical separation condition and contrastive pseudo-word training objective.
3. **Dual-Speed Poison-Resistant Continual Learning (§4):** An external-oracle learning architecture combining sub-millisecond hippocampal recall (`episodic.json`), a monotonic join-semilattice quarantine ledger (`quarantine.json`), deterministic PII scrubbing, held-out benchmark leakage blocking, idle cortical retraining, and an online circuit breaker.
4. **End-to-End Native Rust & Emacs FFI Verification (§5–6):** Lock-free concurrent inference with SHA-256 catalog hot-reloading over the Emacs C module ABI, validated across 2,785 regression cases, 277 blind cases, and 30 real Emacs 31.1 execution fixtures.

2 Typed Open-Action Basis and Zero-`Eval` Safety Boundary

A core design axiom of LispM LLM is that *neural outputs are untrusted proposals over a typed symbolic basis, never executable code.* Even with an open action set, arbitrary S-expressions remain unrepresentable.

Definition 1 (Open Action Specification)**.** *An action specification is a 5-tuple* $s = (\mathrm{id}, f, \kappa, \delta, \mu) \in \mathcal{A}$ *where:*

1. $\mathrm{id} \in \Sigma_{\mathrm{pub}}$ *is a unique action identifier matching* `^[a-z][a-z0-9-]*$` *of length* ≤ 128*, containing no private Emacs separator (`--`) and no trailing hyphen;*
2. $f \in \Sigma_{\mathrm{pub}} \setminus \mathcal{D}$ *is a public Emacs Lisp function symbol outside a permanent denylist* $\mathcal{D}$ *of evaluators, process spawners, and destructive primitives (`eval`, `funcall`, `apply`, `load`, `shell-command`, `call-process`, `start-process`, `delete-file`, `write-region`, `kill-emacs`, etc.);*
3. $\kappa \in \mathcal{K}$ *is a typed argument contract, either* $\mathsf{Fixed}(c_1, \ldots, c_m)$ *with* $m \in \{0, 1, 2\}$ *typed constants* $c_j \in \mathbb{Z} \cup \Sigma^*_{valid}$*, or* $\mathsf{String}(\phi, \Omega)$ *binding exactly one JSON string field* $\phi \neq$ `"action"` *validated against domain* $\Omega \in \{\mathsf{Any}, \mathsf{FolderBasename}, \mathsf{HttpUrl}\}$*;*
4. $\delta \in \Sigma^+$ *is a natural-language action summary encoded by the action tower;*
5. $\mu \in \Sigma_{\mathrm{pub}} \cup \{\bot\}$ *is an optional major-mode guard restricting when* s *is eligible.*

Bounded Clause Decomposition and Identifier Normalization. Given a raw user instruction x, `parse_open_command` first extracts up to two JSON-quoted string literals (v_1, v_2), replacing each literal in-place with the sentinel token `NAME` so literal contents neither consume token budget nor leak into prose syntax checks. Mixed letter–digit identifier tokens outside quotes (such as `1password`, `utf8`, `base64`) expand deterministically into English number words (`one password`), whereas standalone numeric quantifiers (*"move forward 3 words"*) lack alphabetic characters and are rejected syntactically. Negations (`not`, `never`, `don't`), conditionals (`if`, `unless`, `when`), and ambiguous bare `and` connectors are rejected prior to neural scoring; sequential clauses split on `then` / `and then` into $L \in \{1, 2\}$ clauses $(u_1, \ldots, u_L)$. For a catalog of size $M = |\mathcal{A}|$, a valid program $P \in \mathcal{P}(\mathcal{A})$ is either (s_{i_1}) or (s_{i_1}, s_{i_2}) with $i_1 \neq i_2$, yielding $|\mathcal{P}(\mathcal{A})| = M + M(M-1)$ distinct typed programs (2,304 programs at $M = 48$, scaling quadratically without grammar changes).

Origin-Locked Execution Preflight. When a proposal reaches Emacs via the C ABI, `lispm-llm.el` decodes JSON strictly (rejecting duplicate keys or unknown fields), constructs `(cons function args)` data lists without `read`/`eval`, displays a non-intrusive review buffer, and verifies on `C-c C-c` that the originating window, buffer, point, region, narrowing bounds, modification tick, directory, `fboundp` binding, and active `derived-mode-p` hierarchy all remain intact before invoking `(apply fn args)`.

3 Scale-Invariant Dual-Tower Bi-Encoder Architecture

3.1 Shared Word + Subword Transformer Backbone

Both user clauses ($r = 0$) and action descriptors ($r = 1$) pass through a shared lightweight encoder so subword and lexical semantics align in low-resource regimes. Each token w_t at position $t \in \{0, \ldots, T-1\}$ ($T \leq 60$, prepended by $\langle$`cls`$\rangle$) is mapped to a vocabulary index $\mathrm{id}(w_t) \in \{0, \ldots, |V|-1\}$ and up to $G = 32$ hashed character n-grams ($n \in \{3, 4, 5\}$ over boundary-marked $\langle w_t \rangle$) in $B = 2{,}048$ buckets. Let $c_t \in \mathbb{R}^d$ ($d = 64$) denote the variance-stabilized character

feature:

$$c_t = \frac{1}{\sqrt{\max(1, |\{g : h_{t,g} \neq 0\}|)}} \sum_{g=1}^{G} E_{\text{char}}[h_{t,g}], \tag{1}$$

$$h_t^{(0)} = \text{LN}(E_{\text{word}}[\text{id}(w_t)] + c_t + E_{\text{pos}}[t] + E_{\text{role}}[r]). \tag{2}$$

Two Pre-Norm Transformer encoder layers ($H = 4$ heads, $d_{\text{ff}} = 128$, GELU activation, dropout 0.1) transform $H^{(0)} \in \mathbb{R}^{T \times d}$ into $H^{(2)} \in \mathbb{R}^{T \times d}$. Role-specific projections $W_q, W_a \in \mathbb{R}^{d \times d}$ map the $\langle$`cls`$\rangle$ state $h_0^{(2)}$ onto the unit hypersphere $\mathbb{S}^{d-1}$:

$$\hat{q}_{\text{sem}} = \frac{W_q h_0^{(2)} + b_q}{\|W_q h_0^{(2)} + b_q\|_2}, \qquad \hat{a}_{\text{sem}} = \frac{W_a h_0^{(2)} + b_a}{\|W_a h_0^{(2)} + b_a\|_2}. \tag{3}$$

In parallel, we compute a *salience-weighted character-n-gram summary* $\hat{u}_{\text{lex}} \in \mathbb{S}^{d-1}$ directly from $c_{1:T-1}$, down-weighting high-frequency command wrapper stop-words $\mathcal{F}$ (*please, open, show, the, current, buffer, window, ...*) by $\alpha_t = 0.08$ for $w_t \in \mathcal{F}$ and $\alpha_t = 1.0$ for domain-salient words:

$$u_{\text{lex}} = \sum_{t=1}^{T-1} \alpha_t c_t, \qquad \hat{u}_{\text{lex}} = \frac{u_{\text{lex}}}{\max(\|u_{\text{lex}}\|_2, \varepsilon)}. \tag{4}$$

3.2 Why Naive Bi-Encoders Fail on Zero-Shot Open Actions

Suppose query and action representations mix $\hat{q}_{\text{sem}}$ and $\hat{q}_{\text{lex}}$ into a single vector in $\mathbb{R}^d$ scored against both catalog actions and a learnable bank of $K = 16$ unit-norm negative prototypes $\{\hat{r}_k\}_{k=1}^{K} \subset \mathbb{S}^{d-1}$. During training on the 48-action corpus, instructions such as *"open calculator"*, *"open terminal"*, or *"copy current line"* carry ground-truth label `reject`. Consequently, some prototype $\hat{r}_{k^*}$ aligns strongly ($\langle \hat{q}_{\text{sem}}, \hat{r}_{k^*} \rangle \approx 0.85$) with the contextual pattern "open $\langle$`unk`$\rangle$" or "copy $\langle$`unk`$\rangle$". Furthermore, if semantic temperature s_{sem} is unconstrained relative to lexical weight s_{lex} (in our ablation, unconstrained s_{sem} grew to 30.0 while $s_{\text{lex}} \approx 6.6$), even a modest contextual gap $\langle \hat{q}_{\text{sem}}, \hat{a}_{\text{sem}} \rangle - \langle \hat{q}_{\text{sem}}, \hat{r}_{k^*} \rangle = -0.25$ induces a -7.5 logit penalty that drowns out exact subword overlap ($+6.6 \times 0.85 = +5.6$), causing systematic false refusals on newly registered open actions.

3.3 Scale-Invariant Semantic–Lexical Bilinear Decomposition

We resolve this by decoupling the semantic and lexical channels into orthogonal subspaces of $\mathbb{R}^{2d}$ while coupling their scales multiplicatively, and restricting negative prototypes strictly to the semantic subspace. Let

$$s_{\text{sem}} = \exp(\text{clamp}(\ell, \ln 4, \ln 16)), \qquad s_{\text{lex}} = s_{\text{sem}} \cdot (\rho_0 + \text{softplus}(g)), \quad \rho_0 = 0.55, \tag{5}$$

where $\ell, g \in \mathbb{R}$ are learnable scalars. Define the $2d$-dimensional (128-float) query and action embeddings:

$$q_{\text{vec}} = \begin{bmatrix} \sqrt{s_{\text{sem}}}\, \hat{q}_{\text{sem}} \\ \sqrt{s_{\text{lex}}}\, \hat{q}_{\text{lex}} \end{bmatrix} \in \mathbb{R}^{2d}, \qquad a_{\text{vec},i} = \begin{bmatrix} \sqrt{s_{\text{sem}}}\, \hat{a}_{\text{sem},i} \\ \sqrt{s_{\text{lex}}}\, \hat{a}_{\text{lex},i} \end{bmatrix} \in \mathbb{R}^{2d}. \tag{6}$$

Given pre-encoded catalog matrix $A \in \mathbb{R}^{M \times 2d}$ and compatibility mask $M_{\text{compat}}(q, s_i) \in \{0, -10^4\}$ (which masks actions whose string-argument arity disagrees with `NAME` or whose mode $\mu_i \neq \perp$ is absent from the active buffer's `derived-mode-parent` chain), logits over the M actions and class $M + 1$ (`reject`) are:

$$z_i(q; \mathcal{A}) = \langle q_{\text{vec}}, a_{\text{vec},i} \rangle + M_{\text{compat}}(q, s_i) = s_{\text{sem}} \langle \hat{q}_{\text{sem}}, \hat{a}_{\text{sem},i} \rangle + s_{\text{lex}} \langle \hat{q}_{\text{lex}}, \hat{a}_{\text{lex},i} \rangle + M_{\text{compat}}(q, s_i), \tag{7}$$

$$z_{\text{rej}}(q) = s_{\text{sem}} \max_{k \in \{1,\ldots,K\}} \langle \hat{q}_{\text{sem}}, \hat{r}_k \rangle. \tag{8}$$

Proposition 1 (Simultaneous Open-Vocabulary Separation)**.** *Consider an out-of-vocabulary noun phrase ν with contextual semantic cosines $c_{\text{pos}} = \langle \hat{q}_{\text{sem}}(verb^+, \nu), \hat{a}_{\text{sem}}(verb^+, \nu) \rangle$, $c_{\text{neg}} = \langle \hat{q}_{\text{sem}}(verb^-, \nu), \hat{a}_{\text{sem}}(verb^+, \nu) \rangle$ for an inverted/destructive verb $verb^-$, and reject alignment $c_r = \max_k \langle \hat{q}_{\text{sem}}, \hat{r}_k \rangle$. Suppose matched salient subwords satisfy $\langle \hat{q}_{\text{lex}}, \hat{a}_{\text{lex}} \rangle \geq \gamma_{\text{hit}}$ whereas unmatched catalog entries satisfy $\langle \hat{q}_{\text{lex}}, \hat{a}_{\text{lex},j} \rangle \leq \gamma_{\text{miss}}$ with $0 \leq \gamma_{\text{miss}} < \gamma_{\text{hit}} \leq 1$. Whenever the scale-invariant ratio $\rho = s_{\text{lex}}/s_{\text{sem}} \geq \rho_0$ and contextual cosines satisfy*

$$\rho\,\gamma_{\text{miss}} < c_r - c_{\text{pos}} < \rho\,\gamma_{\text{hit}} < c_r - c_{\text{neg}}, \tag{9}$$

a single dot-product router in $\mathbb{R}^{2d}$ simultaneously guarantees:

1. ***Unmatched OOV Refusal:*** *If no action in $\mathcal{A}$ describes ν, $\max_j z_j(q; \mathcal{A}) < z_{\text{rej}}(q)$;*
2. ***Zero-Shot Open Admission:*** *Inserting s_ν into $\mathcal{A}$ yields $z_\nu(q; \mathcal{A} \cup \{s_\nu\}) > z_{\text{rej}}(q)$ without updating any weight;*
3. ***Destructive-Verb Safety:*** *Querying $(verb^-, \nu)$ against $\mathcal{A} \cup \{s_\nu\}$ still yields $z_\nu < z_{\text{rej}}$.*

Proof. Divide $z_i - z_{\text{rej}}$ by $s_{\text{sem}} > 0$: for (1), $(c_{\text{pos}} - c_r) + \rho\gamma_{\text{miss}} < 0$; for (2), $(c_{\text{pos}} - c_r) + \rho\gamma_{\text{hit}} > 0$; and for (3), $(c_{\text{neg}} - c_r) + \rho\gamma_{\text{hit}} < 0$. Because $s_{\text{lex}} = \rho\, s_{\text{sem}}$, all three inequalities are invariant to temperature scaling. □

3.4 Contrastive Pseudo-Word Training and Calibration

To drive parameters directly into the regime of Proposition 1, each training batch combines: (i) stratified general, short (≤ 4 words), and hard examples across catalog actions with 10% word-to-⟨`unk`⟩ dropout on queries and 20% word subsampling on action descriptions; (ii) 48 corpus `reject` examples; and (iii) 12 dynamically synthesized pseudo-word actions s_m^{syn} (8 zero-arg, 4 string-taking with `NAME`) paired with three contrastive queries per synthetic action: a matched verb query $(\text{verb}^+, \nu_m) \mapsto m$, an inverted/destructive verb query $(\text{verb}^-, \nu_m) \mapsto$ `reject` (e.g., *open/delete*, *copy/purge*, *lock/unlock*), and a valid verb on an *unmatched* novel noun $(\text{verb}^+, \tilde{\nu}_m) \mapsto$ `reject`. Checkpoints are evaluated on the 1,182-row calibration split across a threshold grid $(\tau_p, \tau_m) \in [0.50, 0.95] \times [0.00, 0.30]$, maximizing integer decision utility $U = +1 \cdot N_{\text{sup,ok}} - 8 \cdot N_{\text{sup,wrong}} + 2 \cdot N_{\text{unsup,rej}} - 10 \cdot N_{\text{unsup,false}}$ subject to reference non-regression bounds.

4 Dual-Speed Poison-Resistant Continual Learning

Static weights alone cannot anticipate personal abbreviations (e.g., mapping *"summon git porcelain overview"* to `magit-status`). Conversely, updating neural weights on every keystroke is computationally wasteful and vulnerable to label noise. Inspired by complementary learning systems, `lispm-llm-learn` couples a fast hippocampal store with a slow, gated cortical consolidator.

4.1 External-Oracle Supervision Only

Raw model self-predictions are structurally unrepresentable in the observation schema (`OracleSource`). Supervision enters exclusively through two verified external channels:

1. **Behavioral Corrections:** An abnormal lifecycle hook and bounded `post-command-hook` monitor proposals in their originating buffer (strictly ignoring minibuffer input and UI/navigation housekeeping commands). When a proposal is refused, cancelled after preview (`C-c C-k`), or reverted by an immediate undo within $\Delta t_{\text{undo}} = 8\,\text{s}$, the user's subsequent bounded command within $\Delta t_{\text{corr}} = 10\,\text{s}$ is captured as a ground-truth correction.

2. **Teacher Distillation:** When local inference refuses a request and delegates to a higher-tier agent via `lispm-llm-fallback-function`, verified action calls or explicit `'reject` verdicts are recorded through `lispm-llm-learn-record-teacher`.

4.2 Join-Semilattice Quarantine Algebra and Privacy Scrubbing

Before touching disk, every ingested clause passes through three deterministic filters in Rust:

- **PII Literal Scrubbing:** Quoted string literals in `ParsedClause` are replaced at ingestion with canonical dummy constants (`"https://example.org"` for `HttpUrl` contracts, `"example"` for all other string contracts), preserving exact `NAME` clause syntax while guaranteeing zero persistence of private filenames, URLs, or search terms.
- **Held-Out Benchmark & Calibration Shield:** Every clause normalized via NFKC + lowercase tokenization (`clause_key`) is checked against the union of frozen calibration rows, frozen command benchmarks, and real Emacs task instructions; overlaps increment `withheld_skipped` and never enter episodic memory or training (`Withheld`).
- **Monotonic Semilattice Quarantine:** Each normalized clause key k maps to a state in $(\mathcal{S}, \sqcup)$:

$$\begin{aligned}\mathcal{S} = \{\bot\} \cup \{\mathsf{Consistent}(y, n, h) : y \in \mathcal{Y},\, n \geq 1,\, h \in \{\mathsf{false}, \mathsf{true}\}\} \\ \cup \{\mathsf{Conflicted}(L, n) : L \subseteq \mathcal{Y},\, |L| \geq 2\},\end{aligned} \tag{10}$$

  with join operator $\mathsf{Consistent}(y_1, n, h_1) \sqcup \mathsf{Obs}(y_2, h_2) = \mathsf{Consistent}(y_1, n+1, h_1 \vee h_2)$ if $y_1 = y_2$, and $\mathsf{Conflicted}(\{y_1, y_2\}, n+1)$ if $y_1 \neq y_2$, while $\mathsf{Conflicted}(L, n) \sqcup \mathsf{Obs}(y, _) = \mathsf{Conflicted}(L \cup \{y\}, n+1)$.

Theorem 1 (Order-Independent Ambiguity Quarantine)**.** $(\mathcal{S}, \sqcup)$ *is a commutative, associative, idempotent-label join-semilattice in which* Conflicted *is an absorbing upper stratum. Any clause key k that ever receives two distinct labels—whether across separate sessions, within a self-contradictory two-clause observation, or against manual* `examples.json`*—transitions irreversibly to* Conflicted *(unless explicitly reset by an operator* `replace-dataset`*) and is atomically expunged from both* `episodic.json` *and* `examples.json`*.*

4.3 Dual-Speed Recall, Idle Consolidation, and Circuit Breaker

Every Consistent clause materializes atomically into `<workspace>/episodic.json`, consulted in $O(1)$ tree-map lookup *before* neural scoring on the very next command without reloading tensors. During Emacs idle periods (≥ 180 s with ≥ 5 unconsolidated examples), a non-blocking state machine runs `learn-sync` $\rightarrow$ `train` $\rightarrow$ `benchmark` $\rightarrow$ `finalize` $\rightarrow$ `activate`. A candidate run can be finalized *only* if it achieves zero regressions in supported accuracy, wrong executions, supported refusals, unsupported false accepts, and real Emacs task completions relative to the frozen baseline; failed generations are marked `generation_exhausted` so idle timers never spin infinitely. Finally, auto-promoted models are monitored online: if the user cancellation + immediate-undo rate exceeds 35% over ≥ 8 evaluations, the circuit breaker automatically invokes `rollback` to restore the previous verified artifact.

5 Native Systems Architecture

LispM LLM is implemented in three Rust crates (`lispm-llm`, `lispm-llm-native-api`, `lispm-llm-emacs`) backed by LibTorch 2.9.0 via `tch-rs`, plus four lexical-binding Emacs Lisp modules (`lispm-llm.el`, `lispm-llm-catalog.el`, `lispm-llm-learn.el`, `lispm-llm-train.el`).

- **Lock-Free Inference vs. Exclusive Training:** LibTorch's global RNG state is protected by a non-blocking process-wide `RuntimeGuard` acquired *only* during model construction and training. An already-initialized inference worker thread serves predictions, reads atomic `episodic.json` updates, and hot-recompiles `catalog.json` upon SHA-256 change under `tch::no_grad` without acquiring `RuntimeGuard`, keeping interactive editing responsive even while background retraining executes.
- **Live `lispm-kit` Harvesting:** `lispm-llm-catalog-harvest` introspects Emacs's `obarray` for modes declaring `(get mode 'lispm-kit-commands)` (recursively flattening `:submenu` trees and concatenating `:label`, `:button`, `:help`, and docstring summaries under `:mode`), top-level `lispm-*` openers, and curated/registered Emacs commands whose `interactive-form` specifies zero arguments or a single string prompt (`s`, `M`, `f`, `F`, `D`, `b`, `B`).

6 Empirical Evaluation

All experiments were executed locally on macOS Apple Silicon (CPU-only, LibTorch 2.9.0) with SHA-256-bound artifacts verified by `export_hf.py`. Table 1 summarizes the published checkpoints. Training the 264,322-parameter open bi-encoder from scratch for 2,400 steps over 17,232 training rows (selecting step 2,200 on the 1,182-row calibration split at $\tau_p = 0.60, \tau_m = 0.00$) takes **163.0 seconds** on 4 CPU threads. End-to-end evaluation across all 277 blind cases (348 clauses, including process startup, SafeTensors SHA-256 verification, 48-action catalog compilation, and JSON output) completes in **283.6 ms** (**1.02 ms/request** on 2 CPU threads).

Table 1: Published SafeTensors checkpoints in `ElderRyan/lispm-llm`.

Artifact	Architecture	Params	Train Rows	CPU Train	Action Space
`python-reference/` (`.`)	2L Transformer + 49-way Linear	184,625	16,404	—	Closed (48 + `reject`)
`rust-trained/`	2L Transformer + 49-way Linear	184,625	16,404	317.8 s	Closed (48 + `reject`)
`semantic-trained/`	2L Transformer + 49-way Linear	189,425	17,232	128.7 s	Closed (48 + `reject`)
`biencoder/`	2L Dual-Tower ($\mathbb{R}^{128}$) + 16 Rej. Bank	**264,322**	17,232	163.0 s	**Open $\mathcal{A}$ + Mode Mask**

6.1 Frozen 48-Action Regression, Blind Generalization, and Real Emacs Execution

Table 2 compares the historical default (`python-reference`), the closed `semantic-trained` candidate, and the open `biencoder` on identical frozen benchmarks. On real Emacs 31.1 execution fixtures (`data/emacs-tasks.json`, 21 single-step and 9 two-step tasks verified against ground-truth buffer, point, region, window-layout, and filesystem states in isolated child Emacs processes), all three models achieve **30/30 (100%)** goal completion with zero wrong states.

On the independently authored, overlap-filtered blind two-action suite (`blind_composition`, 71 cases), `biencoder` achieves the highest accuracy of all checkpoints (**66/71 = 93.0%**, vs. 54/71 for default and 62/71 for `semantic-trained`), and outperforms the default baseline on all 2,256 deterministic two-action recombinations (**2,050/2,256** vs. 2,028/2,256) as well as novel-to-both cross-domain compositions (**298/352** vs. 269/352). On single-clause supported wording regression (286 cases), `biencoder` cuts semantic wrong executions by 3× (**2** wrong vs. 6 in both closed models) and reduces unsupported regression false accepts from 11 to **7**, trading slightly more conservative refusals on unfamiliar single-word synonyms prior to episodic adaptation.

Table 2: Frozen regression, blind audit, and real Emacs 31.1 execution results. Supported cohorts report **Correct / Wrong / Rejected**; unsupported cohorts report **Rejected / False Accepts**.

Cohort (N)	**Default 48-cls (`.`)**	**Closed `semantic-trained`**	**Open `biencoder`**
Frozen Supported Command Regression & Two-Action Recombinations			
Historical Emacs (187)	173 / 2 / 12	**178** / **1** / 8	175 / 3 / 9
Historical LispM (182)	176 / **0** / 6	175 / **0** / 7	**176** / **0** / 6
Supported wording robustness (286)	266 / 6 / 14	**271** / 6 / 9	256 / **2** / 28
Composition: Emacs × Emacs (552)	474 / 9 / 69	**499** / **4** / 49	485 / 16 / 51
Composition: LispM × LispM (552)	**519** / **0** / 33	514 / **0** / 38	517 / **0** / 35
Composition: Cross-Domain (1,152)	1,035 / 10 / 107	**1,058** / **4** / 90	1,048 / 18 / 86
Independent Blind Generalization Suite (`nov28-blind`, 277 cases)			
Blind single-action supported (118)	107 / **1** / 10	**114** / **1** / 3	100 / **1** / 17
Blind two-action composition (71)	54 / 2 / 15	62 / **0** / 9	**66** / 1 / 4
Out-of-Scope Safety Refusal Cohorts (Rejected / False Accepts)			
Historical Emacs unsupported (56)	**56 / 0**	55 / 1	55 / 1
Historical LispM unsupported (61)	60 / 1	60 / 1	**61 / 0**
Fresh robustness unsupported (126)	115 / 11	**120 / 6**	119 / 7
Blind unsupported (88)	86 / 2	**87 / 1**	84 / 4
Real Isolated Batch Emacs 31.1 State-Execution Suite			
1-step & 2-step goal reachability (30)	**30 / 30 (100%)**	**30 / 30 (100%)**	**30 / 30 (100%)**

6.2 Zero-Shot Open-Action Routing, Mode Masking, and Scale-Invariance Ablation

To evaluate open-vocabulary routing beyond the 48 training actions, we extended `biencoder/catalog.json` at runtime—with *zero gradient updates*—with unseen LispM applications (`lispm-1password`, `lispm-security`, `lispm-shrdlu`), mode-scoped `lispm-kit` commands (`lispm-1password-copy-totp` with μ = `lispm-1password-mode`), and standard Emacs commands (`magit-status`, `rgrep` with `String("pattern", Any)`). Table 3 contrasts the unconstrained-scale ablation ($s_{\text{sem}} \leq 30$, additive $s_{\text{lex}} \approx 6.6$) against our scale-invariant bilinear formulation ($s_{\text{sem}} \leq 16$, $s_{\text{lex}} = s_{\text{sem}}(0.55 + \text{softplus}(g))$):

Table 3: Zero-shot routing and safety behavior on unseen open actions (no retraining).

Instruction	**Active Mode**	**Unconstrained**	**Scale-Invariant (Ours)**
`open the 1password vault`	`fundamental-mode`	`(lispm-1password)`	`(lispm-1password)`
`open the security audit dashboard`	`fundamental-mode`	`(lispm-security)`	`(lispm-security)`
`launch shrdlu blocks world planner`	`fundamental-mode`	*Refused (false rej.)*	`(lispm-shrdlu)`
`open magit git status`	`fundamental-mode`	`(magit-status)`	`(magit-status)`
`recursively grep for "RuntimeGuard"`	`fundamental-mode`	*Refused (false rej.)*	`(rgrep "RuntimeGuard")`
`copy the totp verification code`	`fundamental-mode`	*Refused*	**Refused** (mode-masked)
`copy the totp verification code`	`lispm-1password-mode`	*Refused (false rej.)*	`(lispm-1password-copy-totp)`
`purge the totp verification code`	`lispm-1password-mode`	*Refused*	**Refused** (destructive verb)
`open spotify music player`	`fundamental-mode`	*Refused*	**Refused** (unmatched app)

As predicted by Proposition 1, coupling s_{lex} proportionally to s_{sem} and training with synthetic string-taking and verb-inverted pseudo-words eliminates false rejections on `lispm-shrdlu`, `rgrep`, and in-mode `lispm-1password-copy-totp`, while strictly preserving mode masking outside `lispm-1password-mode`, rejecting destructive verb inversions (*"purge ..."*), and refusing uninstalled applications (*"open spotify ..."*). Furthermore, when an unseen slang phrase has zero lexical overlap with its target descriptor (e.g., *"summon git porcelain overview"* $\mapsto$ `magit-status`), recording a single external observation populates `episodic.json` immediately and resolves subsequent invocations in < 1 ms via the native FFI test suite.

7 Limitations and Conclusion

LispM LLM deliberately bounds its scope to one or two sequential, non-conditional operations with at most one typed string literal per clause. Multi-step planning ($\geq$ 3 actions), loops, conditionals, numeric repeat counts, and multi-argument interactive commands belong to higher-tier deliberative agents (`lispm-agent`), for which LispM LLM serves as the fast, zero-`eval` reflex layer and continual distillation target. Within this reflex layer, combining a scale-invariant semantic–lexical dual-tower router with a semilattice-quarantined episodic/cortical learning loop demonstrates that sub-300K-parameter local models can remain simultaneously fast ($\sim$ 1 ms CPU), open-vocabulary, privacy-preserving, and poison-resistant.

Reproducibility & Artifacts. Full Rust, C ABI, and Emacs Lisp source code, frozen training/calibration/blind/task datasets, SHA-256-verified SafeTensors checkpoints, and release manifests are publicly available under AGPL-3.0-only at `https://huggingface.co/ElderRyan/lispm-llm`.

References

[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "Siamese" time delay neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1993.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146, 2017.

[3] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992, 2019.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, et al. Learning transferable visual models from natural language supervision. In *Proceedings of ICML*, pages 8748–8763, 2021.

[5] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex. *Psychological Review*, 102(3):419–457, 1995.

[6] R. M. Stallman. EMACS: The extensible, customizable self-documenting display editor. In *Proceedings of the ACM SIGPLAN SIGOA Symposium on Text Manipulation*, pages 147–156, 1981.