FOCON: FORMULA-CONDITIONED RECEIVER-AWARE CLAUSE COMMUNICATION NETWORK FOR PARALLEL SAT SOLVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned-clause exchange is a key mechanism in parallel conflict-driven clause learning (CDCL), whereas existing sharing policies largely assess clause utility through global statistics, overlooking the dependence of utility on formula structure and the evolving search state of the receiver. We formulate clause exchange as formula-conditioned, receiver-aware communication and propose FOCON, a lightweight learned router that selects clause imports without modifying symbolic CDCL. A polarity-equivariant factor network processes each CNF once to produce signed structural context and formula-specific routing coefficients. Each candidate transfer is then scored from clause structure and directed sender-receiver states through a compact event-level scorer. FOCON learns from latency-discounted clause reuse, regulates communication through adaptive budget control, and falls back to native sharing outside the learned support. Experiments demonstrate that FOCON achieves aggregate gains over state-of-the-art learning-guided methods and solver baselines while accepting only about 20% of candidate transfers and remaining effective across unseen SAT families and distinct problem encodings.

1 INTRODUCTION

Boolean satisfiability (SAT) provides a computational foundation for formal verification, planning, and automated reasoning. Modern solvers build on CDCL, which converts conflicts into reusable clauses that reshape subsequent search Audemard & Simon (2009). Recent learning-guided methods exploit the structure of conjunctive normal form (CNF) formulas to predict variable phases, select decomposition variables, or parameterize branching heuristics Wang et al. (2024); Zhai & Ge (2025); Tönshoff & Grohe (2026). Parallel CDCL exchanges learned clauses across diversified search trajectories, while regulating communication through receiver-agnostic clause statistics. Logical validity does not ensure computational relevance because the same clause can trigger immediate propagation for one receiver while offering little utility to another in a different search region. Recent neural SAT methods learn variable-level search guidance Wang et al. (2024); Zhai & Ge (2025); Tönshoff & Grohe (2026), whereas learning-augmented exact solvers optimize selected internal decisions Ling et al. (2024); Puigdemont et al. (2024); Deza et al. (2025); Feng & Yang (2025); Reijnen et al. (2025). Clause communication differs from both because transfer utility depends jointly on formula structure, receiver state, and search stage.

This dependence creates a difficult communication trade-off in which permissive sharing increases propagation and database-management overhead, while aggressive rejection can discard clauses that would prune future conflicts. Resolving this trade-off requires predictions at a much higher frequency than branching or solver-level decisions, which rules out repeated graph inference, and relies on indirect supervision because utility becomes observable only through delayed reuse along policy-dependent search trajectories. Structural shifts across formula families compound the challenge by degrading score reliability Ma et al. (2024b); Li et al. (2024b); Yang et al. (2024), motivating a router that combines low-cost event scoring, online traffic regulation, and native fallback outside the calibrated region.

We introduce FOCON, a formula-conditioned and receiver-aware framework for learned-clause routing in parallel CDCL. A polarity-equivariant factor network processes each CNF once to de-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

rive a signed structural context, formula-specific scoring coefficients, and an embedding for support detection, after which each clause-receiver event is scored from the candidate clause and directed worker states through a lightweight inner product. The router learns from latency-discounted reuse, assigning greater utility to clauses reused earlier by the receiver, while a feedback-controlled threshold regulates accepted traffic and an embedding gate restores native sharing beyond the calibrated region. Our contributions are listed as follows:

- We formulate parallel learned-clause sharing as budgeted directed communication, replacing receiver-independent ranking with recipient-specific import decisions. Clause utility is therefore relational, determined by whether a clause produced by the originating worker complements the evolving search trajectory of the receiver within the input formula.
- We propose a formula-conditioned and receiver-aware routing framework that processes each CNF once through a polarity-equivariant factor network and reduces subsequent clause-import decisions to lightweight inner-product scoring. Training uses latency-discounted clause reuse, while a feedback-controlled threshold regulates the acceptance rate and an embedding-based out-of-distribution gate restores native sharing outside the calibrated region.
- Experiments across diverse SAT benchmarks, supported by ablations and theoretical analysis, show that FOCON improves solving efficiency and clause-communication efficiency while retaining native CDCL correctness across unseen formula families and problem encodings.

2 Related Work

Recent neural SAT methods encode CNF structure with graph neural networks and inject variable-level guidance into symbolic search. Existing interfaces include pre-search phase initialization, state-dependent split selection, and one-shot branching parameterization learned from solver feedback, reducing repeated neural inference while retaining CDCL as the reasoning engine Wang et al. (2024); Zhai & Ge (2025); Tönshoff & Grohe (2026). These methods primarily influence variable assignment or decomposition and do not model the receiver-specific value of learned clauses exchanged during parallel search. Beyond SAT, learning-augmented exact solvers optimize selected components of symbolic reasoning, including cut generation and removal Ling et al. (2024); Puigdemont et al. (2024); Deza et al. (2025), branching and model reduction Kuang et al. (2024); Feng & Yang (2025); Liu et al. (2025); Li et al. (2025c), and transferable representations or solver configurations Li et al. (2025a); Huang et al. (2026); Gao et al. (2025); Reijnen et al. (2025); Lu et al. (2025). These approaches demonstrate that learned intervention can improve exact optimization while preserving symbolic inference, but typically attach each learned decision to a single solver state or problem instance rather than to a directed interaction between concurrent search processes. More broadly, neural combinatorial optimization studies transferable representations, diverse solution strategies, scalable decomposition, and test-time or uncertainty-aware optimization Drakulić et al. (2025); Hottung et al. (2025); Chen et al. (2026); Luo et al. (2025); Li et al. (2025b); Zheng et al. (2024); Ouyang et al. (2026); Bi et al. (2026); Verdù et al. (2025); Li et al. (2024c); Ichikawa (2024); Ma et al. (2024a); Xiao et al. (2024); Geng et al. (2024), while neural algorithmic reasoning aligns learned representations with primal–dual, discrete, or retrieval-augmented computation He & Vitercik (2025); Rodionov & Prokhorenkova (2025); Li et al. (2024a). These works focus on solution construction, algorithm emulation, or solver control, leaving high-frequency communication among asynchronous symbolic workers largely unmodeled. Reliable learned intervention further requires recognizing formulas outside the calibrated support. Existing work develops statistically grounded out-of-distribution (OOD) tests, structural extrapolation for graphs, and bounded energy functions under graph distribution shift Ma et al. (2024b); Li et al. (2024b); Yang et al. (2024). However, these methods assess predictive support independently of downstream solver dynamics and therefore do not address when learned communication should abstain in favor of native sharing. FOCON targets this gap by coupling formula-conditioned structure, receiver-dependent clause utility, budgeted communication, and support-aware fallback within parallel CDCL. Further discussion is provided in Appendix A.

3 BACKGROUND

3.1 SAT AND CONFLICT-DRIVEN CLAUSE LEARNING

The SAT problem asks whether a propositional formula admits a satisfying assignment. Let $\mathcal{V} = \{x_1, \ldots, x_n\}$ denote the variable set and $\mathcal{L} = \{x_i, \neg x_i \mid x_i \in \mathcal{V}\}$ the corresponding literal set. A CNF formula with m clauses is

$$F = \bigwedge_{j=1}^{m} C_j, \qquad C_j = \bigvee_{\ell \in \mathcal{L}_j} \ell, \qquad \mathcal{L}_j \subseteq \mathcal{L}. \tag{1}$$

The formula is satisfiable if some assignment $\mathbf{v} \in \{0,1\}^n$ satisfies every clause C_j and unsatisfiable otherwise. CDCL extends Boolean constraint propagation (BCP) and branching with conflict analysis, clause learning, and non-chronological backtracking Biere et al. (2009). When propagation exposes a conflict, analysis of the implication graph derives a learned clause C^{L} and returns the search to a level where C^{L} becomes asserting. Soundness ensures $F \models C^{\mathrm{L}}$, and therefore $\mathrm{Mod}(F \wedge C^{\mathrm{L}}) = \mathrm{Mod}(F)$, where $\mathrm{Mod}(F)$ denotes the set of satisfying assignments of F. Learned clauses therefore prevent recurrence of the analyzed conflict without excluding any satisfying assignment. Modern CDCL solvers further employ efficient propagation data structures, restart policies, dynamic branching, and learned-clause database reduction Biere et al. (2009); Tönshoff & Grohe (2026). Clause length and literal block distance (LBD) provide efficient criteria for clause retention and export Audemard & Simon (2009; 2017), but encode clause-level quality rather than relevance to a particular receiving search trajectory.

3.2 PARALLEL CDCL AND LEARNED-CLAUSE EXCHANGE

Learned clauses are logically sound; however, logical validity does not necessarily determine computational benefit. Parallel CDCL makes this distinction operational, because multiple workers explore different search regions and repeatedly exchange learned clauses. A parallel CDCL solver assigns the shared formula to a worker set $\mathcal{W} = \{1, \ldots, w\}$, each worker $w \in \mathcal{W}$ maintains a private search state $\mathbf{s}_w$ and learned-clause database $\mathcal{C}_w^{\mathrm{L}}$. Search diversification drives workers toward distinct conflict regions, while clause exchange carries derived consequences between their trajectories Audemard & Simon (2017). The computational value of an import therefore depends on the receiver state, since the same clause can trigger propagation and prevent future conflicts for one worker but remain satisfied or redundant for another, as illustrated in Figure 1. Receiver-independent criteria, including clause length and literal block distance, abstract away this context, so permissive policies increase propagation and database overhead, whereas restrictive policies discard receiver-relevant conflict information. Clause-level quality therefore does not fully characterize communication utility.

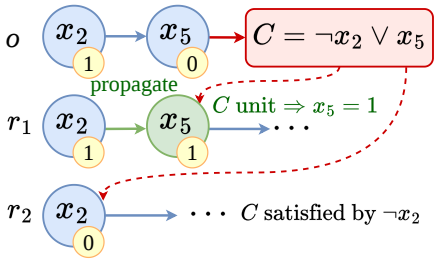

Figure 1: Receiver-dependent clause utility. The origin worker o derives $C = \neg x_2 \vee x_5$ and transfers the clause to receivers r_1 and r_2. The clause becomes unit and forces $x_5 = 1$ at r_1, whereas $\neg x_2$ already satisfies the clause at r_2

3.3 BUDGETED RECEIVER-AWARE CLAUSE ROUTING

We formulate learned-clause exchange as directed routing over solver states. A candidate event and the corresponding routing decision are $e = (F, C_e, o_e, r_e, \mathbf{s}_{o_e}, \mathbf{s}_{r_e})$ and $a_e = \pi(e) \in \{0,1\}$, where C_e is the candidate clause, $o_e, r_e \in \mathcal{W}$ denote the origin and receiver workers, and $\mathbf{s}_{o_e}, \mathbf{s}_{r_e} \in \mathcal{S}$ denote their current search states. The directed action $a_e \in \{0,1\}$ respectively rejects or imports C_e into r_e, with reversed worker roles defining a distinct event. The utility of event e cannot be observed when the routing decision is made. Let $u_e^\star$ denote the latent contribution of the import to subsequent search. Exact attribution requires paired executions differing only in a_e, but a single transfer can redirect the solver trajectory and shared-worker timing, making the resulting counterfactuals costly and schedule-sensitive. Learning therefore uses an observable event-level surrogate, while end-to-end solving time remains the evaluation criterion.

Let $\mathcal{E}_F(\pi)$ be the policy-dependent set of candidate events encountered while solving F, with $n_F(\pi) = |\mathcal{E}_F(\pi)| > 0$. The realized acceptance rate is $\bar{a}_F(\pi) = \sum_{e \in \mathcal{E}_F(\pi)} a_e / n_F(\pi)$. Given a formula distribution $\mathcal{D}$, target rate $\rho \in (0, 1)$, and tolerance $\varepsilon_\rho > 0$, routing is formulated as

$$\min_{\pi} \ \mathbb{E}_{F \sim \mathcal{D}} \left[t_{\text{e2e}}(F; \pi) \right] \qquad \text{s.t.} \qquad |\bar{a}_F(\pi) - \rho| \leq \varepsilon_\rho, \tag{2}$$

where t_{e2e} includes learned guidance and native solving. The constraint controls transfer frequency rather than direct resource usage, and $\mathcal{E}_F(\pi)$ remains endogenous because accepted clauses reshape subsequent search. Effective routing therefore requires low-cost online decisions conditioned on the formula structure and receiver state.

4 FOCON Framework

Existing clause-sharing policies do not capture how transfer utility varies with formula structure and the evolving state of each receiver, while repeated neural inference is impractical at clause-exchange frequency. These limitations motivate a routing mechanism that combines formula-level structural reasoning with lightweight receiver-aware decisions. FOCON formulates learned-clause exchange as directed and formula-conditioned routing. As illustrated in Figure 2, a polarity-equivariant factor network analyzes each CNF once, encoding signed structure into a compact formula-specific scoring rule and a representation of calibration support. During parallel search, each candidate clause transfer from an origin worker to a receiver is evaluated using their ordered search states, allowing transfer utility to vary across workers and search stages. This separation amortizes graph reasoning over the complete solve and reduces each online decision to lightweight native scoring. Latency-discounted reuse provides supervision, while feedback-controlled thresholding regulates communication. A support-aware gate enables learned routing within the calibrated formula region and restores native sharing for unsupported formulas. Learning governs clause transfer without altering CDCL deduction or proof validity.

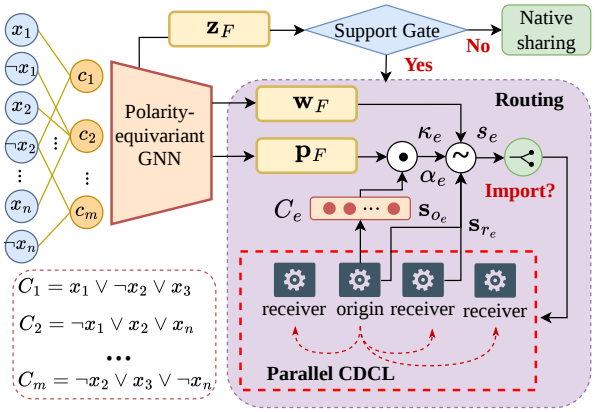

Figure 2: Overview of FOCON. The formula graph is processed once to obtain $\mathbf{p}_F$, $\mathbf{w}_F$, and $\mathbf{z}_F$. The support gate selects between learned routing and native sharing. When activated, the router combines the candidate clause with directed origin-receiver states and scores each transfer through a formula-specific inner product

4.1 Polarity-Equivariant Formula Conditioning

Clause utility depends on the CNF structure underlying search; therefore, FOCON processes each formula once to derive a routing context reused across candidate transfers, avoiding repeated graph inference during parallel search. For a CNF formula F, we construct a bipartite factor graph $\mathcal{G}_F = (\mathcal{V}_L \cup \mathcal{V}_C, \mathcal{E}_F^{\text{fac}})$, where $\mathcal{V}_L$ contains paired nodes $(x_i, \neg x_i)$ for each variable, $\mathcal{V}_C$ contains the original clauses, and $(\ell, C_j) \in \mathcal{E}_F^{\text{fac}}$ whenever $\ell \in C_j$. Identity-free local descriptors capture polarity-sensitive incidence structure and relative clause scale, from which separate encoders initialize $\mathbf{h}_\ell^{(0)}, \mathbf{g}_j^{(0)} \in \mathbb{R}^d$ with latent dimension d. At propagation step $k \in \{0, \ldots, K-1\}$, each clause aggregates the representations of incident literals, after which each literal receives the updated clause context together with the representation of the complementary literal as follows

$$\mathbf{m}_j^{(k)} = \text{mean}_{\ell \in C_j} \mathbf{h}_\ell^{(k)}, \qquad \mathbf{g}_j^{(k+1)} = \mathbf{g}_j^{(k)} + f_C^{\text{upd}} \left([\mathbf{g}_j^{(k)}; \mathbf{m}_j^{(k)}] \right), \tag{3}$$

$$\mathbf{n}_\ell^{(k)} = \text{mean}_{j : \ell \in C_j} \mathbf{g}_j^{(k+1)}, \qquad \mathbf{h}_\ell^{(k+1)} = \mathbf{h}_\ell^{(k)} + f_L^{\text{upd}} \left([\mathbf{h}_\ell^{(k)}; \mathbf{h}_{\bar{\ell}}^{(k)}; \mathbf{n}_\ell^{(k)}] \right), \tag{4}$$

where K denotes the propagation depth and $\bar{\ell}$ the complement of ℓ, and $f_C^{\mathrm{upd}} : \mathbb{R}^{2d} \to \mathbb{R}^d$ and $f_L^{\mathrm{upd}} : \mathbb{R}^{3d} \to \mathbb{R}^d$ are learnable update maps shared across propagation steps. Therefore, the paired embeddings define an antisymmetric orientation and a symmetric magnitude for each variable, which are calculated as follows

$$u_i = f_{\mathrm{pol}}(\mathbf{h}_{x_i}^{(K)}) - f_{\mathrm{pol}}(\mathbf{h}_{\neg x_i}^{(K)}), \qquad v_i = f_{\mathrm{mag}}\left(\frac{\mathbf{h}_{x_i}^{(K)} + \mathbf{h}_{\neg x_i}^{(K)}}{2}\right), \tag{5}$$

$$p_i = \tanh(u_i)\tanh(|v_i|), \qquad \mathbf{p}_F = [p_1, \ldots, p_n]^\top \in [-1, 1]^n. \tag{6}$$

Here, $f_{\mathrm{pol}} : \mathbb{R}^d \to \mathbb{R}$ and $f_{\mathrm{mag}} : \mathbb{R}^d \to \mathbb{R}$ are shared scalar-valued heads, p_i is the signed structural coordinate of variable x_i, and $\mathbf{p}_F$ denotes the structural context of formula F. A polarity reversal $x_i \leftrightarrow \neg x_i$ maps p_i to $-p_i$, enabling learned clauses to be compared with the formula under the same symmetry, which coordinates represent structural orientation. A permutation-invariant aggregation operator $\mathcal{A}$ combines the final literal and clause embeddings, while a compact formula-scale descriptor $\mathbf{x}_F$ preserves global CNF statistics. The formula representation is

$$\mathbf{h}_F = f_F^{\mathrm{enc}}\left(\left[\mathcal{A}\left(\{\mathbf{h}_\ell^{(K)}\}_{\ell \in \mathcal{V}_L}, \{\mathbf{g}_j^{(K)}\}_{C_j \in \mathcal{V}_C}\right); \mathbf{x}_F\right]\right) \in \mathbb{R}^{d_F}, \tag{7}$$

where f_F^{enc} is a learnable formula encoder with output dimension d_F and $\mathbf{x}_F$ denotes the formula-scale descriptor. Let $f_{\mathrm{route}} : \mathbb{R}^{d_F} \to \mathbb{R}^{d_\phi}$ and $f_{\mathrm{sup}} : \mathbb{R}^{d_F} \to \mathbb{R}^q$ be the routing and support heads, respectively, which yeilds

$$\mathbf{w}_F = \frac{f_{\mathrm{route}}(\mathbf{h}_F)}{\sqrt{d_\phi}}, \qquad \mathbf{z}_F = \frac{f_{\mathrm{sup}}(\mathbf{h}_F)}{\max(\|f_{\mathrm{sup}}(\mathbf{h}_F)\|_2, \varepsilon)}. \tag{8}$$

Here, $\mathbf{w}_F \in \mathbb{R}^{d_\phi}$ parameterizes the event scorer, d_ϕ is the event-feature dimension, $\mathbf{z}_F \in \mathbb{R}^q$ is the normalized support embedding, and $\varepsilon > 0$ ensures numerical stability. Thus, one formula-level graph evaluation provides the signed context $\mathbf{p}_F$, the coefficients $\mathbf{w}_F$ governing subsequent routing, and the support representation $\mathbf{z}_F$ used for abstention.

4.2 Directed Receiver-Aware Routing

Formula structure characterizes potentially informative clause patterns, whereas transfer utility depends on the evolving search context of the receiver. For a candidate event e, let $i(\ell)$ index the variable associated with literal ℓ and $\chi(\ell) \in \{-1, 1\}$ encode literal polarity. The alignment between C_e and the signed formula context is summarized by

$$\kappa_e = \frac{1}{|C_e|}\sum_{\ell \in C_e} |p_{i(\ell)}|, \qquad \alpha_e = \frac{1}{|C_e|}\sum_{\ell \in C_e} \chi(\ell) p_{i(\ell)}. \tag{9}$$

Here, κ_e measures structural salience, and α_e retains polarity alignment. Each worker w exposes a compact state vector $\mathbf{s}_w$ that summarizes current search progress, assignment density, and prior communication. Let $\boldsymbol{\phi}_e = [1; \log(1 + |C_e|); \kappa_e; \alpha_e; \mathbf{s}_{o_e}; \mathbf{s}_{r_e}; \mathbf{s}_{r_e} - \mathbf{s}_{o_e}; \kappa_e \mathbf{s}_{r_e}] \in \mathbb{R}^{d_\phi}$ denote the event representation that combines clause context with the ordered worker states. The state difference retains communication direction, whereas $\kappa_e \mathbf{s}_{r_e}$ conditions structural relevance on the current receiver state. Formula-conditioned routing then becomes $s_e = \mathbf{w}_F^\top \boldsymbol{\phi}_e$ and $\widehat{p}_e = \sigma(s_e)$, where σ is the sigmoid function, s_e is the routing logit, and $\widehat{p}_e$ is the corresponding training score. Excluding worker identifiers makes the rule equivariant to any joint relabeling of workers and states. Formula-level graph inference is amortized across all transfers, leaving clause aggregation and a fixed-dimensional inner product for each event.

4.3 Learning from Delayed Solver Feedback

Routing utility is unobserved at decision time; FOCON therefore learns from native solver traces collected before filtering. Let t_e denote the receiver conflict index at event e, and let t_e^+ denote the first later index at which the associated clause is reused in conflict analysis. With $\Delta_e = t_e^+ - t_e$, the target is $y_e = \exp(-\Delta_e/\tau_u)$ for observed reuse and $y_e = 0$ otherwise, where $\tau_u > 0$ controls temporal discounting. Earlier reuse receives greater utility, although y_e remains a surrogate rather than a causal estimate of runtime reduction. Trace termination right-censors late events, prompting

a horizon-dependent weight that discounts negative labels with limited follow-up. Weighted binary cross-entropy losses are combined through smoothed worst-domain aggregation to limit dominance by frequent formula families. For learnable parameters Θ, training minimizes

$$\mathcal{J}(\Theta) = \mathcal{J}_{\mathrm{rob}} + \lambda_b \mathcal{J}_{\mathrm{budget}} + \lambda_i \mathcal{J}_{\mathrm{inv}} + \lambda_a \mathcal{J}_{\mathrm{ac}}, \tag{10}$$

where $\mathcal{J}_{\mathrm{rob}}$ denotes the robust prediction loss, $\mathcal{J}_{\mathrm{budget}}$ aligns score mass with target acceptance rate ρ, and $\mathcal{J}_{\mathrm{inv}}$ and $\mathcal{J}_{\mathrm{ac}}$ enforce domain invariance and prevent representation collapse. The nonnegative coefficients λ_b, λ_i, and λ_a control the corresponding terms. Censor correction and class balancing alter empirical prevalence, so $\widehat{p}_e$ serves as a utility score for ranking.

4.4 Support-Aware Budget Control

Neural routing is activated within the formula region represented by calibration data. Let $\boldsymbol{\mu}, \mathbf{v} \in \mathbb{R}^q$ denote the coordinate-wise mean and variance of validation embeddings. The standardized support distance is $d_{\mathrm{sup}}(F) = \sqrt{\left(\sum_{j=1}^{q} (z_{Fj} - \mu_j)^2 / v_j\right)/q}$. The learned router operates when $d_{\mathrm{sup}}(F) \leq \tau_{\mathrm{ood}}$; outside this region, clause exchange follows the native policy. The gate confines neural intervention to the calibrated support and provides no formal out-of-distribution guarantee. Short clauses retain native acceptance, while candidates with $|C_e| > 2$ receive $a_e = 1$ when $s_e \geq \theta_e$ and $a_e = 0$ otherwise. After each event, the controller updates $\theta_{e+1} = \mathrm{clip}_{[\theta_{\min}, \theta_{\max}]}(\theta_e + \eta(a_e - \rho))$, where $\eta > 0$ is the controller step size and $[\theta_{\min}, \theta_{\max}]$ bounds the score threshold. This negative feedback tracks target acceptance rate ρ, subject to mandatory short-clause transfers. After a single formula-level graph evaluation, each routing decision requires only clause aggregation and a d_ϕ-dimensional inner product, yielding $O(|C_e| + d_\phi)$ cost. The router selects which logically entailed clauses are communicated, while all deductive operations and proof validation remain under native CDCL.

5 Theoretical Analysis

Let $\Gamma = (\sigma_V, \sigma_C, \boldsymbol{\epsilon}, \sigma_W)$ denote a joint transformation of variables, clauses, literal polarities, and workers, where $\boldsymbol{\epsilon} \in \{-1, 1\}^n$ encodes polarity reversals. Let $\mathbf{P}_{\sigma_V}$ and $\mathbf{S}_{\boldsymbol{\epsilon}}$ denote the corresponding variable permutation and polarity matrices.

Proposition 1. *Assume shared literal and clause maps and permutation-invariant formula aggregation. For any transformed formula F_Γ and event e_Γ, $\mathbf{p}_{F_\Gamma} = \mathbf{P}_{\sigma_V}\mathbf{S}_{\boldsymbol{\epsilon}}\mathbf{p}_F$, $\mathbf{w}_{F_\Gamma} = \mathbf{w}_F$, $\mathbf{z}_{F_\Gamma} = \mathbf{z}_F$, and $s_{e_\Gamma} = s_e$. Thus, routing is invariant to variable, clause, polarity, and worker relabeling, while exchanging origin and receiver can change the score.*

To isolate the value of receiver state, let ρ_{short} denote the expected rate of mandatory short-clause transfers and assume $\rho_{\mathrm{short}} < \rho < 1$. Conditioning on $|C_e| > 2$, define the residual budget $\rho_{\mathrm{long}} = (\rho - \rho_{\mathrm{short}})/(1 - \rho_{\mathrm{short}})$. Let $\xi = (C_e, \mathbf{s}_{o_e})$, $q = \mathbb{E}[y_e \mid F, \xi, \mathbf{s}_{r_e}]$, and $\bar{q} = \mathbb{E}[q \mid F, \xi]$. Under the same budget, let $\varphi_0 = \varphi_0(F, \xi)$ denote a receiver-agnostic acceptance rule and $\varphi_1 = \varphi_1(F, \xi, \mathbf{s}_{r_e})$ a receiver-aware rule, and define $j_0 = \sup_{\varphi_0 : \mathbb{E}[\varphi_0] = \rho_{\mathrm{long}}} \mathbb{E}[q\varphi_0]$ and $j_1 = \sup_{\varphi_1 : \mathbb{E}[\varphi_1] = \rho_{\mathrm{long}}} \mathbb{E}[q\varphi_1]$.

Theorem 1. *Receiver conditioning satisfies $j_1 \geq j_0$. Moreover, letting θ^* denote an optimal receiver-aware acceptance threshold, $j_1 > j_0$ whenever a positive-probability set of contexts ξ satisfies $\Pr(q < \theta^* \mid F, \xi) > 0$ and $\Pr(q > \theta^* \mid F, \xi) > 0$.*

Theorem 1 motivates an explicit structure-receiver interaction. Partition the formula-conditioned coefficient vector according to the blocks of $\boldsymbol{\phi}_e$ as $\mathbf{w}_F = [w_{F,0}; w_{F,\mathrm{len}}; w_{F,\kappa}; w_{F,\alpha}; \mathbf{w}_{F,o}; \mathbf{w}_{F,r}; \mathbf{w}_{F,\Delta}; \mathbf{w}_{F,I}]$. For fixed F and origin state $\mathbf{s}_o$, the scorer reduces to $s_F(C, \mathbf{s}_r) = b_F(C) + \mathbf{a}_F^\top \mathbf{s}_r + \kappa_C \mathbf{c}_F^\top \mathbf{s}_r$, where $b_F(C)$ is receiver-independent, $\mathbf{a}_F = \mathbf{w}_{F,r} + \mathbf{w}_{F,\Delta}$, and $\mathbf{c}_F = \mathbf{w}_{F,I}$.

Proposition 2. *For clauses C, D and receiver states $\mathbf{s}_r, \mathbf{s}_{r'}$, $[s_F(C, \mathbf{s}_r) - s_F(D, \mathbf{s}_r)] - [s_F(C, \mathbf{s}_{r'}) - s_F(D, \mathbf{s}_{r'})] = (\kappa_C - \kappa_D)\mathbf{c}_F^\top(\mathbf{s}_r - \mathbf{s}_{r'})$. Hence, removing $\kappa_e \mathbf{s}_{r_e}$ makes pairwise clause ordering receiver-independent. Conversely, if $\kappa_{C_0} \neq \kappa_{C_1}$ and $\mathbf{s}_0 \neq \mathbf{s}_1$, the scorer can realize opposite preferences across the two receivers, whereas any additive form $b(C) + h(\mathbf{s}_r)$ cannot.*

We next characterize ranking quality under a fixed communication budget. For a realized event window $\mathcal{E}_N \subseteq \mathcal{E}_F(\pi)$, let $q_e = \mathbb{E}[y_e \mid F, C_e, \mathbf{s}_{o_e}, \mathbf{s}_{r_e}]$, $\mathcal{M}_N = \{e \in \mathcal{E}_N : |C_e| \leq 2\}$, $b_N = \lfloor \rho N \rfloor$,

and $k_N = b_N - |\mathcal{M}_N|$, assuming $|\mathcal{M}_N| \le b_N$. Let $\mathcal{S}_N^*$ maximize total q_e among b_N-event sets containing $\mathcal{M}_N$. Let $\widehat{\mathcal{R}}_N$ contain the k_N highest-scoring events in $\mathcal{E}_N \setminus \mathcal{M}_N$, and define $\widehat{\mathcal{S}}_N = \mathcal{M}_N \cup \widehat{\mathcal{R}}_N$. Define $\varepsilon_N^{\mathrm{rank}} = \max_{e,e' \in \mathcal{E}_N \setminus \mathcal{M}_N} |(\widehat{p}_e - \widehat{p}_{e'}) - (q_e - q_{e'})|$.

Theorem 2. *For every realized window with $|\mathcal{M}_N| \le b_N$, $\sum_{e \in \mathcal{S}_N^*} q_e - \sum_{e \in \widehat{\mathcal{S}}_N} q_e \le k_N \varepsilon_N^{\mathrm{rank}}$.*

We finally analyze the deployed threshold controller. Let $m_e = 1$ for mandatory short-clause events and $m_e = 0$ otherwise, and define $\psi_e(\theta) = \rho\theta + m_e(s_e - \theta) + (1 - m_e)(s_e - \theta)_+$. For the fractional allocation relaxation with $a_e \in [m_e, 1]$ and $\sum_e a_e = \rho N$, $\sum_e \psi_e(\theta)$ is the dual objective of score-maximizing allocation. At deployment, $g_e = \rho - a_e \in \partial\psi_e(\theta_e)$ and $|g_e| \le g_\rho := \max\{\rho, 1 - \rho\}$, so the threshold update is projected subgradient descent.

Theorem 3 (Dynamic dual regret of projected control)**.** *Let $\mathcal{I} = [\theta_{\min}, \theta_{\max}]$, $d_\theta = \theta_{\max} - \theta_{\min}$, and $v_N = \sum_{e=0}^{N-2} |\vartheta_{e+1} - \vartheta_e|$ for any comparator sequence $\vartheta_e \in \mathcal{I}$. The update $\theta_{e+1} = \mathrm{clip}_{\mathcal{I}}(\theta_e + \eta(a_e - \rho))$ satisfies $\sum_{e=0}^{N-1} [\psi_e(\theta_e) - \psi_e(\vartheta_e)] \le (d_\theta^2 + 2d_\theta v_N)/(2\eta) + \eta N g_\rho^2/2$. The bound holds for any realized event stream, including streams induced by previous routing decisions, and permits active clipping.*

Corollary 1. *If clipping is inactive for N decisions, then $|\bar{a}_N - \rho| \le (\theta_{\max} - \theta_{\min})/(\eta N)$ and $\bar{a}_N \ge |\mathcal{M}_N|/N$, where $\bar{a}_N = N^{-1} \sum_{e=0}^{N-1} a_e$.*

For a solve of formula F with $N = n_F(\pi)$ routing events, $\bar{a}_N = \bar{a}_F(\pi)$. Hence, when clipping is inactive and the mandatory-transfer floor is feasible, the communication constraint in Equation 2 is satisfied whenever $N \ge (\theta_{\max} - \theta_{\min})/(\eta\varepsilon_\rho)$. Thus, projected control admits dynamic regret on endogenous event streams and tracks the target rate at $O(N^{-1})$ before saturation, subject to the mandatory-transfer floor.

6 Experiments

6.1 Experimental Setting

Benchmarks. The router is trained on 3400 CNF formulas from three benchmark families: 1600 random 3-SAT, 1200 graph-coloring, and 600 cryptographic instances, following Tönshoff & Grohe (2026). We first evaluate in-distribution generalization on 1700 held-out formulas from the same families, including 800 random 3-SAT, 600 graph-coloring, and 300 cryptographic instances. We then evaluate distribution shift on 100 unseen instances drawn from SATLIB[1], covering AIM, Beijing, Dubois, Hanoi, and SSA. Finally, we assess cross-problem transfer on 600 instances derived from Dominating Set, Clique, and Coloring encodings following Interian & Bernardini (2023). Detailed benchmark descriptions are provided in Appendix D.1.

Baselines. We compare FOCON against several state-of-the-art and established SAT baselines spanning CDCL, parallel solving, learning-guided search, and local search: Kissat Biere et al. (2022), Glucose Audemard & Simon (2009), Glucose-Syrup Audemard & Simon (2017), RLAF Tönshoff & Grohe (2026), NeuroBack Wang et al. (2024), and WalkSAT Tönshoff & Grohe (2026). Glucose-Syrup serves as the principal matched-backend baseline because both methods use eight parallel workers, while FOCON replaces native learned-clause sharing with receiver-aware routing. RLAF and NeuroBack represent state-of-the-art learning-guided SAT methods, whereas Kissat, Glucose, and WalkSAT provide strong conventional baselines across complementary solving paradigms. Performance is reported as mean $\pm$ standard deviation over three independent solver runs per formula. Additional implementation details are given in Appendix D.2.

Metrics. We evaluate all methods using solve rate (SR), end-to-end runtime (RT), and PAR-2. For N solver runs, let $s_i \in \{0, 1\}$ indicate whether run i is successfully solved and t_i denote the end-to-end runtime. Solve rate is $\mathrm{SR} = N^{-1} \sum_{i=1}^{N} s_i$, while RT measures the complete wall-clock cost of each run, including neural inference where applicable. With timeout $T_{\max} = 60$s, $\mathrm{PAR\text{-}2} = N^{-1} \sum_{i=1}^{N} [s_i t_i + (1 - s_i) 2T_{\max}]$. PAR-2 jointly reflects runtime and robustness by penalizing unsolved runs, avoiding the optimistic bias of averaging runtime only over solved instances.

[1] https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html

6.2 EXPERIMENTAL RESULTS

Table 1: In-distribution SAT evaluation of FOCON against baseline solvers on the full benchmark

Dataset	Metric	Kissat	Glucose	Glucose-Syrup	WalkSAT	NeuroBack	RLAF	FOCON
Overall	SR (%) ↑	82.16 ± 0.03	72.73 ± 0.03	84.35 ± 0.41	21.55 ± 0.15	81.55 ± 0.09	83.75 ± 0.03	$\mathbf{85.06 \pm 0.18}$
	RT (s) ↓	14.64 ± 0.05	21.64 ± 0.01	14.99 ± 0.23	48.44 ± 0.14	18.07 ± 0.07	16.81 ± 0.00	$\mathbf{14.25 \pm 0.04}$
	PAR-2 (s) ↓	25.34 ± 0.05	37.87 ± 0.02	24.37 ± 0.48	95.51 ± 0.22	29.11 ± 0.10	26.22 ± 0.02	$\mathbf{23.20 \pm 0.14}$
3-SAT	SR (%) ↑	$\mathbf{76.33 \pm 0.07}$	59.04 ± 0.07	71.92 ± 0.44	45.79 ± 0.31	76.29 ± 0.07	73.63 ± 0.00	73.04 ± 0.31
	RT (s) ↓	$\mathbf{17.22 \pm 0.07}$	29.62 ± 0.01	22.75 ± 0.38	35.44 ± 0.29	19.65 ± 0.07	22.89 ± 0.00	21.71 ± 0.14
	PAR-2 (s) ↓	$\mathbf{31.41 \pm 0.11}$	54.00 ± 0.04	39.58 ± 0.64	67.96 ± 0.47	33.84 ± 0.08	38.17 ± 0.00	37.86 ± 0.32
Coloring	SR (%) ↑	92.06 ± 0.10	87.17 ± 0.00	94.33 ± 0.44	0.00 ± 0.00	91.94 ± 0.42	91.78 ± 0.10	$\mathbf{94.50 \pm 0.29}$
	RT (s) ↓	9.21 ± 0.10	12.38 ± 0.01	7.35 ± 0.19	60.00 ± 0.00	12.49 ± 0.16	10.18 ± 0.00	$\mathbf{6.90 \pm 0.05}$
	PAR-2 (s) ↓	13.97 ± 0.09	20.03 ± 0.00	10.74 ± 0.45	120.00 ± 0.00	17.31 ± 0.40	14.94 ± 0.06	$\mathbf{10.19 \pm 0.22}$
Crypto	SR (%) ↑	77.89 ± 0.19	80.33 ± 0.00	97.56 ± 0.84	0.00 ± 0.00	74.78 ± 0.19	94.67 ± 0.00	$\mathbf{98.22 \pm 0.51}$
	RT (s) ↓	18.64 ± 0.00	18.88 ± 0.01	9.60 ± 0.25	60.00 ± 0.00	25.02 ± 0.04	13.86 ± 0.01	$\mathbf{9.08 \pm 0.23}$
	PAR-2 (s) ↓	31.89 ± 0.12	30.54 ± 0.00	11.06 ± 0.73	120.00 ± 0.00	40.11 ± 0.10	16.94 ± 0.01	$\mathbf{10.14 \pm 0.53}$

Table 1 demonstrates that FOCON achieves the strongest aggregate performance on the in-distribution benchmark, with the highest SR and the lowest RT and PAR-2. Against Glucose-Syrup, FOCON increases SR from 84.35% to 85.06% while reducing RT and PAR-2 by 4.9% and 4.8%, respectively, indicating that the PAR-2 gain is accompanied by both broader solve coverage and lower solving cost. FOCON also outperforms RLAF and NeuroBack in aggregate SR and PAR-2, showing that learned clause communication can complement or surpass guidance applied directly to the search process under the evaluated setting. Family-level results further show consistent gains over native sharing: SR increases from 71.92% to 73.04% on 3-SAT, from 94.33% to 94.50% on Coloring, and from 97.56% to 98.22% on Crypto, while PAR-2 decreases by 4.3%, 5.1%, and 8.3%, respectively. The strongest gains arise on the structured Coloring and Crypto families, where FOCON attains the best SR and runtime-based metrics, whereas Kissat remains strongest on 3-SAT. These results suggest that receiver-aware clause routing is most beneficial when formula structure and cross-worker search states provide informative signals for selective clause reuse.

Table 2: Generalization across unseen SAT families and combinatorial encodings

Regime	Family	NeuroBack	RLAF	FOCON
OOD	Beijing	8.08 ± 0.24	1.46 ± 0.34	$\mathbf{0.49 \pm 0.07}$
	AIM	7.13 ± 0.01	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.02 \pm 0.00}$
	Dubois	15.17 ± 0.06	$\mathbf{8.59 \pm 0.00}$	$\mathbf{8.59 \pm 0.00}$
	Hanoi	7.67 ± 0.09	1.31 ± 0.01	$\mathbf{0.07 \pm 0.02}$
	SSA	7.85 ± 0.07	$\mathbf{0.03 \pm 0.00}$	0.03 ± 0.01
Combinatorial	Dominating Set	2.86 ± 0.01	3.13 ± 0.01	$\mathbf{2.80 \pm 0.01}$
	Clique	2.71 ± 0.00	2.96 ± 0.00	$\mathbf{2.66 \pm 0.00}$
	Coloring	3.46 ± 0.01	3.87 ± 0.01	$\mathbf{3.40 \pm 0.01}$

Table 2 evaluates generalization beyond the training distribution. On unseen SAT families, FOCON matches or improves RLAF across all five families while preserving the same solve coverage: all methods attain 100% SR except on Dubois, where each method reaches 92.86%. The largest reductions occur on Beijing and Hanoi, where FOCON decreases PAR-2 from 1.46 to 0.49s and from 1.31 to 0.07s, respectively, while remaining comparable on AIM, Dubois, and SSA. The combinatorial evaluation introduces a distinct change in problem encoding. All methods solve every instance, while FOCON attains the lowest PAR-2 on Dominating Set, Clique, and Coloring. Together, the results show that the learned routing rule extends beyond the training families and remains effective across both unseen SAT families and distinct combinatorial encodings.

Table 3 decomposes performance by SAT/UNSAT status and reports communication behavior for the two parallel-sharing methods. Relative to Glucose-Syrup, FOCON improves both subsets, with the strongest gain on UNSAT formulas: SR increases from 98.45% to 98.99% and PAR-2 decreases by 10.5%. FOCON accepts only 19.95% of candidate transfers, reducing accepted communication from 721.68K to 152.09K while retaining nearly the same number of recorded clause reuses. These results demonstrate that the runtime gain is associated with substantially more selective communication rather than increased clause traffic. Kissat and NeuroBack remain stronger on SAT formulas, whereas FOCON achieves the strongest UNSAT performance, suggesting that receiver-aware routing becomes particularly valuable during longer conflict-driven searches where imported clauses have greater opportunity for repeated downstream reuse. Communication counters are available only for Glucose-Syrup and FOCON, which share the same parallel CDCL backend.

Table 3: SAT/UNSAT performance and learned-clause communication

	SAT		UNSAT		Clause Communication		
Solver	**SR (%) ↑**	**PAR-2 (s) ↓**	**SR (%) ↑**	**PAR-2 (s) ↓**	**Accept. (%)**	**Accepted**	**Reuses**
Kissat	99.40 ± 0.15	$\mathbf{2.38 \pm 0.19}$	85.44 ± 0.07	23.94 ± 0.05	–	–	–
Glucose-Syrup	88.17 ± 0.77	17.65 ± 1.12	98.45 ± 0.37	10.23 ± 0.27	100.00	721.68K	3.16M
NeuroBack	$\mathbf{99.70 \pm 0.15}$	5.45 ± 0.26	84.01 ± 0.18	29.02 ± 0.12	–	–	–
RLAF	90.53 ± 0.00	15.76 ± 0.00	95.43 ± 0.07	15.36 ± 0.04	–	–	–
FOCON	89.27 ± 0.48	16.06 ± 0.42	$\mathbf{98.99 \pm 0.24}$	$\mathbf{9.16 \pm 0.17}$	**19.95**	**152.09K**	3.12M

Table 4 demonstrates complementary contributions from formula conditioning, receiver state, support-aware fallback, and adaptive thresholding. Here, Accept. denotes the percentage of candidate clause-transfer events accepted by the routing policy. Removing the formula prior produces the largest degradation among the learned routing features on the in-distribution benchmark, while removing receiver state causes a smaller but consistent loss, supporting the separation between formula-level structural context and event-level receiver conditioning. Disabling the OOD gate primarily affects shifted formulas, reducing SR and increasing PAR-2, consistent with the intended role of native fallback outside the calibrated support. Fixed thresholding causes the largest overall in-distribution degradation, whereas the full model improves over native sharing while accepting only about one fifth of candidate transfers.

Table 4: Routing-component ablation on in-distribution and OOD benchmarks

	ID			OOD		
Variant	**SR ↑**	**PAR-2 ↓**	**Accept.**	**SR ↑**	**PAR-2 ↓**	**Accept.**
Full	$\mathbf{85.37 \pm 0.48}$	$\mathbf{22.90 \pm 0.56}$	19.73	99.00 ± 0.00	1.24 ± 0.01	22.07
w/o state	85.16 ± 0.30	23.11 ± 0.27	19.95	99.00 ± 0.00	1.24 ± 0.01	20.68
w/o prior	84.67 ± 0.24	23.86 ± 0.18	19.97	99.00 ± 0.00	1.24 ± 0.01	21.91
w/o OOD gate	85.10 ± 0.44	23.34 ± 0.30	19.95	98.67 ± 0.58	1.65 ± 0.69	22.16
Fixed threshold	81.06 ± 0.27	27.33 ± 0.34	12.76	99.00 ± 0.00	1.24 ± 0.01	35.91
Native sharing	84.90 ± 0.32	23.69 ± 0.29	100.00	99.00 ± 0.00	$\mathbf{1.22 \pm 0.01}$	100.00

Additional analyses in Appendix E provide complementary evidence for these gains. Appendix E.1 confirms consistent improvements over native Glucose-Syrup across all three in-distribution families, with larger gains on the structured Coloring and Crypto instances. Appendix E.2 reports detailed solve coverage and runtime across unseen SAT families and distinct combinatorial encodings, further characterizing the observed transfer behavior. Appendix E.3 demonstrates that adaptive control closely tracks target acceptance rates from 5% to 40% and improves SR and PAR-2 over a fixed threshold across all tested budgets. Appendix E.4 reveals that support-aware fallback preserves learned routing for structurally supported formulas and increasingly defers to native sharing under larger distribution shifts. The paired outcomes in Appendix E.5 indicate that FOCON expands solve coverage within the timeout rather than merely accelerating already solved instances. Finally, Appendix E.6 establishes that the advantage over native sharing emerges once nontrivial search develops and persists throughout later search.

7 Conclusion and Limitations

We introduced FOCON, a formula-conditioned, receiver-aware framework that recasts learned-clause exchange in parallel CDCL as selective communication between evolving search processes. By extracting formula structure once and conditioning subsequent transfers on directed sender–receiver context, FOCON enables lightweight routing while preserving native CDCL deduction. Experiments across diverse SAT benchmarks demonstrate improved solving and communication efficiency over the matched parallel backend, with consistent gains across benchmark families and effective transfer to unseen SAT families and distinct combinatorial encodings. The current supervision based on latency-discounted clause reuse remains a surrogate for the causal contribution of an import to solver progress. Future work can develop counterfactual or online utility attribution to align routing decisions more directly with end-to-end solving efficiency.

AI USE STATEMENT

No generative AI tools were used in the research, analysis, or preparation of this manuscript.

ETHICS STATEMENT

The authors have read and adhered to the ICLR Code of Ethics. This work studies learned-clause communication in SAT solving using synthetic or publicly available benchmark instances and involves no human subjects, personal information, or sensitive data. The proposed method modifies clause communication without changing the logical validity of native CDCL reasoning. We are not aware of additional ethical concerns specific to the experiments or methodology presented in this work.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce the proposed method and reported evaluation throughout the paper. Section 4 specifies the routing framework and deployment procedure, while Section 5 and Appendix C provide the theoretical results and complete proofs. Appendix B details the representations, training objective, calibration, and computational complexity, and Appendix D documents benchmark construction, baseline configurations, implementation settings, and the evaluation protocol used for all reported experiments.

REFERENCES

Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern sat solvers. In *IJCAI*, volume 9, pp. 399–404, 2009.

Gilles Audemard and Laurent Simon. Glucose and syrup in the sat'17. *Proceedings of SAT Competition*, pp. 16–17, 2017.

Jieyi Bi, Zhiguang Cao, Jianan Zhou, Wen Song, Yaoxin Wu, Jie Zhang, Yining Ma, and Cathy Wu. Towards efficient constraint handling in neural solvers for routing problems. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 11907–11936, 2026.

Armin Biere, Hans van Maaren, and Toby Walsh. *Handbook of satisfiability*. SAGE Publications Limited, 2009.

Armin Biere, Mathias Fleury, T Balyo, M Heule, and M Iser. Gimsatul, isasat and kissat entering the sat competition 2022. *Proc. of SAT Competition*, pp. 10–11, 2022.

Lvda Chen, Yang Li, and Junchi Yan. MaskCO: Masked generation drives effective representation learning and exploiting for combinatorial optimization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Arnaud Deza, Elias B Khalil, Zhenan Fan, Zirui Zhou, and Yong Zhang. Learn2aggregate: Supervised generation of chvátal-gomory cuts using graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26947–26954, 2025.

Darko Drakulić, Sofia Michel, and Jean-Marc Andreoli. Goal: A generalist combinatorial optimization agent learner. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 52465–52488, 2025.

Shengyu Feng and Yiming Yang. Sorrel: suboptimal-demonstration-guided reinforcement learning for learning to branch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11212–11220, 2025.

Chengrui Gao, Haopu Shang, Ke Xue, and Chao Qian. Neural solver selection for combinatorial optimization. In *Forty-second International Conference on Machine Learning*, 2025.

Haoyu Geng, Hang Ruan, Runzhong Wang, Yang Li, Yang Wang, Lei Chen, and Junchi Yan. Benchmarking pto and pno methods in the predictive combinatorial optimization regime. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 65944–65971. Curran Associates, Inc., 2024. doi: 10.52202/079017-2108.

Yu He and Ellen Vitercik. Primal-dual neural algorithmic reasoning. In *Forty-second International Conference on Machine Learning*, 2025.

André Hottung, Mridul Mahajan, and Kevin Tierney. Polynet: Learning diverse solution strategies for neural combinatorial optimization. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 80417–80435, 2025.

Peixin Huang, Yaoxin Wu, Yining Ma, Cathy Wu, Wei Zhang, and Wen Song. A general neural backbone for mixed-integer linear optimization via dual attention. In *Forty-third International Conference on Machine Learning*, 2026.

Yuma Ichikawa. Controlling continuous relaxation for combinatorial optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 47189–47216. Curran Associates, Inc., 2024. doi: 10.52202/079017-1495.

Yannet Interian and Sara Bernardini. Learning Interpretable Heuristics for WalkSAT. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 365–373, 8 2023. doi: 10.24963/kr.2023/36. URL https://doi.org/10.24963/kr.2023/36.

Yufei Kuang, Jie Wang, Yuyan Zhou, Xijun Li, Fangzhou Zhu, Jianye HAO, and Feng Wu. Towards general algorithm discovery for combinatorial optimization: Learning symbolic branching policy from bipartite graph. In *Forty-first International Conference on Machine Learning*, 2024.

Hefei Li, Chao Peng, Chenyang Xu, and Zhengfeng Yang. Open-book neural algorithmic reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Sirui Li, Janardhan Kulkarni, Ishai Menache, Cathy Wu, and Beibin Li. Towards foundation models for mixed integer linear programming. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 88590–88638, 2025a.

Sirui Li, Wenbin Ouyang, Yining Ma, and Cathy Wu. Learning-guided rolling horizon optimization for long-horizon flexible job-shop scheduling. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 1724–1768, 2025b.

Xiner Li, Shurui Gui, Youzhi Luo, and Shuiwang Ji. Graph structure extrapolation for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning*, 2024b.

Yang Li, Jinpei Guo, Runzhong Wang, Hongyuan Zha, and Junchi Yan. Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c.

Yixuan Li, Can Chen, Jiajun Li, Jiahui Duan, Xiongwei Han, Tao Zhong, Vincent Chau, Weiwei Wu, and Wanyuan Wang. Fast and interpretable mixed-integer linear program solving by learning model reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27090–27098, 2025c.

Haotian Ling, Zhihai Wang, and Jie Wang. Learning to stop cut generation for efficient mixed-integer linear programming. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 20759–20767, 2024.

Haoyang Liu, Jie Wang, Zijie Geng, Xijun Li, Yuxuan Zong, Fangzhou Zhu, Jianye HAO, and Feng Wu. Apollo-milp: An alternating prediction-correction neural solving framework for mixed-integer linear programming. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 94513–94540, 2025.

Chen Lu, Ke Xue, Lei Yuan, Yao Wang, Yaoyuan Wang, Sheng Fu, and Chao Qian. Sequential multi-agent dynamic algorithm configuration. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 14976–15006. Curran Associates, Inc., 2025. doi: 10.52202/085713-0504.

Fu Luo, Xi Lin, Yaoxin Wu, Zhenkun Wang, Tong Xialiang, Mingxuan Yuan, and Qingfu Zhang. Boosting neural combinatorial optimization for large-scale vehicle routing problems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 14754–14780, 2025.

Hengyuan Ma, Wenlian Lu, and Jianfeng Feng. Efficient combinatorial optimization via heat diffusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Xinsong Ma, Xin Zou, and Weiwei Liu. A provable decision rule for out-of-distribution detection. In *Forty-first International Conference on Machine Learning*, 2024b.

Wenbin Ouyang, Sirui Li, Yining Ma, and Cathy Wu. Learning to segment for vehicle routing problems. In *The Fourteenth International Conference on Learning Representations*, 2026.

Pol Puigdemont, Stratis Skoulakis, Grigorios Chrysos, and Volkan Cevher. Learning to remove cuts in integer linear programming. In *Forty-first International Conference on Machine Learning*, 2024.

Robbert Reijnen, Yaoxin Wu, Zaharah Bukhsh, and Yingqian Zhang. Graph-supported dynamic algorithm configuration for multi-objective combinatorial optimization. In *Forty-second International Conference on Machine Learning*, 2025.

Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning. In *Forty-second International Conference on Machine Learning*, 2025.

Jan Tönshoff and Martin Grohe. Learning from algorithm feedback: One-shot SAT solver guidance with GNNs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Federico Julian Camerota Verdù, Lorenzo Castelli, and Luca Bortolussi. Scaling combinatorial optimization neural improvement heuristics with online search and adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27135–27143, 2025.

Wenxi Wang, Yang Hu, Mohit Tiwari, Sarfraz Khurshid, Kenneth McMillan, and Risto Miikkulainen. Neuroback: Improving CDCL SAT solving using graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Pei Xiao, Zizhen Zhang, Jinbiao Chen, Jiahai Wang, and Zhenzhen Zhang. Neural combinatorial optimization for robust routing problem with uncertain travel times. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Shenzhi Yang, Bin Liang, An Liu, Lin Gui, Xingkai Yao, and Xiaofang Zhang. Bounded and uniform energy-based out-of-distribution detection for graphs. In *Forty-first International Conference on Machine Learning*, 2024.

Shumao Zhai and Ning Ge. Learning splitting heuristics in divide-and-conquer sat solvers with reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 99159–99180, 2025.

Zhi Zheng, Changliang Zhou, Xialiang Tong, Mingxuan Yuan, and Zhenkun Wang. Udc: A unified neural divide-and-conquer framework for large-scale combinatorial optimization problems. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 6081–6125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0197.

A Appendix: Additional Related Work

Learning-guided SAT solving. Neural guidance has entered SAT solving through several variable-level interfaces. NeuroBack predicts phases associated with satisfying assignments and evaluates the graph network once before CDCL begins, thereby avoiding repeated neural inference during search Wang et al. (2024). RDC-SAT represents the current divide-and-conquer state as a graph and trains an actor-critic policy to select split variables Zhai & Ge (2025). RLAF instead predicts variable weights and polarities from the input formula, using solver cost as reinforcement-learning feedback for one-shot branching guidance Tönshoff & Grohe (2026). These approaches differ in supervision and inference schedule, but all influence how search assigns or decomposes variables. None of these decision interfaces evaluates whether a learned clause derived by one worker is relevant to the evolving trajectory of another.

Learning within exact combinatorial solvers. Learning-augmented MILP and ILP solvers provide a broader account of selective intervention within symbolic optimization. HYGRO learns when further cut generation no longer justifies the associated cost, while learned cut-removal policies identify constraints whose continued presence impedes optimization Ling et al. (2024); Puigdemont et al. (2024). Learn2Aggregate predicts which constraints should participate in Chvátal–Gomory aggregation Deza et al. (2025). At the branching level, GS4CO discovers interpretable symbolic policies from bipartite problem graphs, whereas SORREL combines offline reinforcement learning and self-imitation to exploit suboptimal demonstrations Kuang et al. (2024); Feng & Yang (2025). Apollo-MILP alternates prediction with trust-region correction before fixing variables, and learned model reduction constructs smaller formulations rather than predicting complete solutions Liu et al. (2025); Li et al. (2025c). Other methods pursue transferable MILP representations, reusable neural backbones, solver selection, and dynamic configuration Li et al. (2025a); Huang et al. (2026); Gao et al. (2025); Reijnen et al. (2025); Lu et al. (2025). Although these methods preserve symbolic optimization, their learned decisions concern a single solver or instance and do not represent communication utility as a directed relation between concurrent states.

Structured neural optimization and algorithmic reasoning. Generalist and multi-strategy neural solvers share representations across problem classes or solution behaviors, while masked pre-training seeks reusable structural features Drakulić et al. (2025); Hottung et al. (2025); Chen et al. (2026). Scaling methods decompose large instances through learned partitioning, segmentation, rolling horizons, or local improvement Luo et al. (2025); Li et al. (2025b); Zheng et al. (2024); Ouyang et al. (2026); Bi et al. (2026); Verdù et al. (2025). Complementary test-time methods refine latent solutions through objective-guided search, continuous-relaxation annealing, or heat-diffusion transformations Li et al. (2024c); Ichikawa (2024); Ma et al. (2024a). Robust and predictive optimization further study uncertainty-aware construction and the downstream effect of prediction errors Xiao et al. (2024); Geng et al. (2024). Neural algorithmic reasoning takes a more structured route by aligning message passing with primal–dual procedures, constraining execution to discrete states, or retrieving training instances during reasoning He & Vitercik (2025); Rodionov & Prokhorenkova (2025); Li et al. (2024a). Despite their different mechanisms, these methods allocate neural computation to solution construction or algorithm emulation rather than repeated communication decisions whose value changes across receivers and search stages.

Out-of-distribution detection and reliable deployment. Distribution shift is particularly consequential when neural predictions intervene in a symbolic solver. General OOD research derives decision rules from hypothesis testing and controls detection errors under distributional assumptions Ma et al. (2024b). Graph OOD generalization instead extrapolates graph structure beyond the training support, whereas energy-based detection limits extreme scores and logit shifts in graph neural networks Li et al. (2024b); Yang et al. (2024). SAT formulas introduce related shifts through variable scale, clause composition, incidence structure, and generating family. Existing OOD methods provide useful detection principles, but a low OOD score does not establish that neural intervention reduces solving time, and a high score does not identify the appropriate native fallback. Relating formula-level uncertainty to communication policy and end-to-end solver cost therefore remains unresolved.

B Appendix: Additional Methodological Details

This appendix specifies the feature construction, delayed-reuse objective, calibration procedure, deployment rule, and computational complexity underlying Section 4.

B.1 Formula and Event Representations

For $\mathcal{G}_F = (\mathcal{V}_L \cup \mathcal{V}_C, \mathcal{E}_F^{\mathrm{fac}})$, let d_ℓ denote the degree of literal ℓ, $d_i = d_{x_i} + d_{\neg x_i}$ the aggregate degree of variable x_i, and $b_i = |d_{x_i} - d_{\neg x_i}| / \max(d_i, 1)$ the polarity imbalance. The literal descriptor is

$$\mathbf{x}_\ell = \left[\log(1+d_\ell);\, \log(1+d_i);\, b_i;\, \frac{d_\ell}{\max(d_i,1)}\right] \in \mathbb{R}^4. \tag{11}$$

For clause C_j, let $k_j = |C_j|$ and $\delta_{j,s} = 1$ when $k_j = s$ and $\delta_{j,s} = 0$ otherwise. The clause descriptor is

$$\mathbf{x}_{C_j} = \left[\log(1+k_j);\, \frac{k_j}{n};\, \delta_{j,2};\, \delta_{j,3}\right] \in \mathbb{R}^4, \tag{12}$$

where n denotes the number of variables. With m clauses, let $\iota_F = \sum_{j=1}^m k_j$ denote total literal incidence and $r_s = m^{-1}\sum_{j=1}^m \delta_{j,s}$ the proportion of clauses of size s. Formula scale is represented by

$$\mathbf{x}_F = \left[\log(1+n);\, \log(1+m);\, \log\!\left(1+\frac{m}{n}\right);\, \log\!\left(1+\frac{\iota_F}{n}\right);\, r_2;\, r_3\right] \in \mathbb{R}^6. \tag{13}$$

The node encoders initialize $\mathbf{h}_\ell^{(0)} = f_L^{\mathrm{enc}}(\mathbf{x}_\ell)$ and $\mathbf{g}_j^{(0)} = f_C^{\mathrm{enc}}(\mathbf{x}_{C_j})$. The permutation-invariant aggregation in Equation 7 uses mean and max pooling, with formula-scale features appended to the pooled representation:

$$\mathbf{h}_F^{\mathrm{pool}} = \left[\mathrm{mean}_{\ell\in\mathcal{V}_L}\, \mathbf{h}_\ell^{(K)};\, \max_{\ell\in\mathcal{V}_L}\, \mathbf{h}_\ell^{(K)};\, \mathrm{mean}_{C_j\in\mathcal{V}_C}\, \mathbf{g}_j^{(K)};\, \max_{C_j\in\mathcal{V}_C}\, \mathbf{g}_j^{(K)};\, \mathbf{x}_F\right], \tag{14}$$

$$\mathbf{h}_F = f_F^{\mathrm{enc}}\left(\mathbf{h}_F^{\mathrm{pool}}\right). \tag{15}$$

The support head is instantiated as $f_{\mathrm{sup}}(\mathbf{h}_F) = \mathbf{B}_z\mathbf{h}_F + \mathbf{b}_z$, where $\mathbf{B}_z \in \mathbb{R}^{q\times d_F}$ and $\mathbf{b}_z \in \mathbb{R}^q$.

For worker w, let n_w^{conf}, n_w^{dec}, n_w^{prop}, n_w^{rst}, n_w^{asg}, and n_w^{imp} denote cumulative conflicts, decisions, propagations, restarts, assigned variables, and imported clauses, respectively. Define $b_\gamma(x) = \min\{1, \log(1+x)/\gamma\}$. The worker state is

$$\mathbf{s}_w = \left[b_{\gamma_c}(n_w^{\mathrm{conf}});\, b_{\gamma_d}(n_w^{\mathrm{dec}});\, b_{\gamma_p}(n_w^{\mathrm{prop}});\, b_{\gamma_r}(n_w^{\mathrm{rst}});\, \min\!\left(1, \frac{n_w^{\mathrm{asg}}}{n}\right);\, b_{\gamma_i}(n_w^{\mathrm{imp}})\right] \in [0,1]^6. \tag{16}$$

Together with the scalar clause features, two worker states, their directed difference, and the receiver interaction, Equation 9 gives $d_\phi = 28$.

B.2 Delayed-Reuse Objective

Solver termination right-censors events without observed reuse. Let $t_{r_e}^{\mathrm{end}}$ denote the final conflict index observed for receiver r_e and $h_e = \max(0, t_{r_e}^{\mathrm{end}} - t_e)$ the remaining observation horizon. The censor-aware weight is

$$\omega_e = \begin{cases} 1, & y_e > 0, \\ \omega_{\min} + (1-\omega_{\min})\min\!\left(1, \dfrac{h_e}{\tau_u}\right), & y_e = 0, \end{cases} \tag{17}$$

where $\omega_{\min} \in (0,1)$ limits negative evidence from short continuations.

For a mini-batch, let n_+ and n_- denote the numbers of events with positive and zero targets. The positive-class weight is

$$\beta_+ = \mathrm{clip}_{[1,\beta_{\max}]}\left(\frac{n_-}{\max(n_+,1)}\right), \tag{18}$$

and the censor-aware event loss is

$$\mathcal{J}_e = -\omega_e \left[\beta_+ y_e \log \widehat{p}_e + (1 - y_e) \log(1 - \widehat{p}_e)\right]. \tag{19}$$

Let $\mathcal{D}_B$ denote the formula domains represented in the mini-batch and $\mathcal{E}_g$ the events from domain $g \in \mathcal{D}_B$. Define

$$\mathcal{J}_g = \frac{1}{|\mathcal{E}_g|} \sum_{e \in \mathcal{E}_g} \mathcal{J}_e. \tag{20}$$

The smoothed worst-domain objective is

$$\mathcal{J}_{\text{rob}} = \tau_{\text{rob}} \left[\log \sum_{g \in \mathcal{D}_B} \exp\left(\frac{\mathcal{J}_g}{\tau_{\text{rob}}}\right) - \log |\mathcal{D}_B|\right], \tag{21}$$

where $\tau_{\text{rob}} > 0$ controls concentration on high-loss domains.

For the batch event set $\mathcal{E}_B$, define the mean routing score

$$\bar{p}_B = \frac{1}{|\mathcal{E}_B|} \sum_{e \in \mathcal{E}_B} \widehat{p}_e, \tag{22}$$

with budget-shaping loss

$$\mathcal{J}_{\text{budget}} = (\bar{p}_B - \rho)^2. \tag{23}$$

This term shapes the score distribution during training, whereas realized communication is regulated by the online threshold controller.

Let $\mathcal{F}_g$ contain formula embeddings from domain $g \in \mathcal{D}_B$, and define

$$\bar{\mathbf{z}}_g = \frac{1}{|\mathcal{F}_g|} \sum_{F \in \mathcal{F}_g} \mathbf{z}_F, \qquad \bar{\mathbf{z}} = \frac{1}{|\mathcal{D}_B|} \sum_{g \in \mathcal{D}_B} \bar{\mathbf{z}}_g. \tag{24}$$

Domain alignment is encouraged by

$$\mathcal{J}_{\text{inv}} = \frac{1}{q|\mathcal{D}_B|} \sum_{g \in \mathcal{D}_B} \|\bar{\mathbf{z}}_g - \bar{\mathbf{z}}\|_2^2. \tag{25}$$

For the batch formula set $\mathcal{F}_B$, let

$$\bar{z}_{Bj} = \frac{1}{|\mathcal{F}_B|} \sum_{F \in \mathcal{F}_B} z_{Fj}, \qquad \nu_j = \sqrt{\frac{1}{|\mathcal{F}_B|} \sum_{F \in \mathcal{F}_B} (z_{Fj} - \bar{z}_{Bj})^2 + \varepsilon_{\text{ac}}}. \tag{26}$$

The anti-collapse term is

$$\mathcal{J}_{\text{ac}} = \frac{1}{q} \sum_{j=1}^{q} [\nu_{\min} - \nu_j]_+^2, \tag{27}$$

where $[x]_+ = \max(x, 0)$, $\varepsilon_{\text{ac}} > 0$ stabilizes the dispersion estimate, and $\nu_{\min} > 0$ defines the minimum coordinate variation. Together, $\mathcal{J}_{\text{rob}}$, $\mathcal{J}_{\text{budget}}$, $\mathcal{J}_{\text{inv}}$, and $\mathcal{J}_{\text{ac}}$ form Equation 10. Gradients update the factor network and output heads without differentiating through CDCL.

B.3 Calibration and Deployment

Let $\mathcal{T} \subset (0, 1)$ contain candidate score thresholds obtained from a validation grid and empirical quantiles. For $\tau \in \mathcal{T}$, let $a_{\text{val}}(\tau)$ denote the validation acceptance rate under the mandatory short-clause rule and $f_{1,g}(\tau)$ the reuse F1 on domain g. The initial score threshold is selected as

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} \left[\min_g f_{1,g}(\tau) - \lambda_\rho \left|a_{\text{val}}(\tau) - \rho\right|\right]. \tag{28}$$

Because deployment thresholds the routing logit s_e, the initial controller threshold is

$$\widetilde{\tau}^* = \text{clip}_{[\varepsilon_t, 1-\varepsilon_t]}(\tau^*), \qquad \theta_0 = \log \frac{\widetilde{\tau}^*}{1 - \widetilde{\tau}^*}. \tag{29}$$

Algorithm 1 Support-aware clause routing with FOCON

Require: CNF formula F, parameters Θ, target rate ρ, controller step η, thresholds $\theta_0, \theta_{\min}, \theta_{\max}, \tau_{\text{ood}}$, and support statistics $(\boldsymbol{\mu}, \mathbf{v})$
Ensure: Native CDCL outcome

```
Construct G_F and compute (p_F, w_F, z_F)
Compute d_sup(F)
if d_sup(F) > τ_ood then
    Run parallel CDCL with native clause sharing
else
    Initialize θ ← θ_0
    for each candidate routing event e do
        Read C_e, o_e, r_e, s_{o_e}, and s_{r_e}
        Compute κ_e, α_e, and φ_e
        s_e ← w_F^⊤ φ_e
        if |C_e| ≤ 2 or s_e ≥ θ then
            a_e ← 1 and import C_e into receiver r_e
        else
            a_e ← 0
        end if
        θ ← clip_{[θ_min, θ_max]}(θ + η(a_e − ρ))
    end for
end if
return the outcome produced by native CDCL
```

This conversion places θ_0 on the logit scale without interpreting $\widehat{p}_e$ as a calibrated probability.

Let $\mathcal{F}_{\text{val}}$ denote the validation formulas used for support calibration. For each coordinate j,

$$\mu_j = \frac{1}{|\mathcal{F}_{\text{val}}|} \sum_{F \in \mathcal{F}_{\text{val}}} z_{Fj}, \qquad v_j = \max\left\{ \varepsilon_s,\ \frac{1}{|\mathcal{F}_{\text{val}}|} \sum_{F \in \mathcal{F}_{\text{val}}} (z_{Fj} - \mu_j)^2 \right\}, \tag{30}$$

where $\varepsilon_s > 0$ provides a variance floor. The support threshold τ_{ood} is selected from a validation quantile of $d_{\text{sup}}(F)$. If $d_{\text{sup}}(F) > \tau_{\text{ood}}$, the solver uses native clause sharing for the complete run.

Algorithm 1 summarizes deployment. Budget tracking of the threshold update is analyzed in Theorem 3 and Corollary 1.

B.4 Computational Complexity

The symmetry, receiver-conditioning, logical-preservation, ranking, and budget properties are established in Appendix C. We therefore focus here on the computational cost of the deployed router.

For latent dimension d and K propagation steps, formula-level message passing requires

$$O\big(K\,[|\mathcal{E}_F^{\text{fac}}|d + (2n+m)d^2]\big) \tag{31}$$

computation and

$$O\big(|\mathcal{E}_F^{\text{fac}}| + (2n+m)d\big) \tag{32}$$

storage. This cost is incurred once per formula. Each subsequent routing event requires

$$O(|C_e| + d_\phi) \tag{33}$$

computation for clause-context aggregation and the formula-conditioned inner product, without additional graph-network evaluation.

C Appendix: Additional Theory and Proofs

This appendix proves Proposition 1, Theorem 1, Proposition 2, Theorem 2, Theorem 3, and Corollary 1.

C.1 Proof of Proposition 1

Proof. Variable and clause permutations only reindex the corresponding factor nodes without changing graph incidence. Since the literal and clause encoders, update maps, and aggregation operators are shared, the node representations transform consistently under σ_V and σ_C at every propagation step. This follows directly by induction on k: the correspondence holds at initialization, and mean aggregation commutes with node permutations while the same update maps are applied to corresponding nodes.

A polarity reversal exchanges the paired literal representations associated with each reversed variable. Consequently, the antisymmetric component changes sign while the symmetric component is preserved. For variable x_i,

$$(\mathbf{p}_{F_\Gamma})_{\sigma_V(i)} = \epsilon_i(\mathbf{p}_F)_i,$$

and therefore

$$\mathbf{p}_{F_\Gamma} = \mathbf{P}_{\sigma_V}\mathbf{S}_{\boldsymbol{\epsilon}}\mathbf{p}_F.$$

Permutation-invariant formula aggregation removes node ordering, giving

$$\mathbf{w}_{F_\Gamma} = \mathbf{w}_F, \qquad \mathbf{z}_{F_\Gamma} = \mathbf{z}_F.$$

Let ℓ_Γ denote the transformed literal corresponding to ℓ. Under a consistent polarity reversal, $\chi(\ell_\Gamma) = \epsilon_i\chi(\ell)$ and $(\mathbf{p}_{F_\Gamma})_{\sigma_V(i)} = \epsilon_i(\mathbf{p}_F)_i$. Hence,

$$\left|(\mathbf{p}_{F_\Gamma})_{\sigma_V(i)}\right| = \left|(\mathbf{p}_F)_i\right|, \qquad \chi(\ell_\Gamma)(\mathbf{p}_{F_\Gamma})_{\sigma_V(i)} = \chi(\ell)(\mathbf{p}_F)_i,$$

which preserves κ_e and α_e. Worker relabeling through σ_W carries the ordered origin–receiver state pair to the corresponding transformed event, so $\boldsymbol{\phi}_{e_\Gamma} = \boldsymbol{\phi}_e$ and $s_{e_\Gamma} = s_e$. Exchanging the origin and receiver roles instead changes $\mathbf{s}_{r_e} - \mathbf{s}_{o_e}$ and $\kappa_e\mathbf{s}_{r_e}$ in general, preserving communication direction.

Finally, let $\mathcal{C}_w^{\mathrm{imp}}$ denote the clauses imported by worker w. Since every candidate learned clause satisfies $F \models C_e$,

$$\mathrm{Mod}\left(F \wedge \bigwedge_{C\in\mathcal{C}_w^{\mathrm{imp}}} C\right) = \mathrm{Mod}(F).$$

Accepting a transfer therefore adds only a logically entailed clause, whereas rejecting a transfer leaves the local formula unchanged. Any routing sequence thus preserves the model set and native CDCL correctness. □

C.2 Proof of Theorem 1

Proof. We first establish a budgeted threshold identity. For any random variable $Q \in [0,1]$ and acceptance rule $A \in [0,1]$ satisfying $\mathbb{E}[A] = \rho_{\mathrm{long}}$, every $\theta \in [0,1]$ gives

$$\mathbb{E}[QA] = \theta\rho_{\mathrm{long}} + \mathbb{E}[(Q-\theta)A] \le \theta\rho_{\mathrm{long}} + \mathbb{E}[(Q-\theta)_+].$$

A threshold rule that accepts above θ, rejects below θ, and randomizes at equality attains the bound for a threshold satisfying

$$\Pr(Q > \theta) \le \rho_{\mathrm{long}} \le \Pr(Q \ge \theta).$$

Hence,

$$\sup_{A:\mathbb{E}[A]=\rho_{\mathrm{long}}} \mathbb{E}[QA] = \min_{\theta\in[0,1]} \left\{\theta\rho_{\mathrm{long}} + \mathbb{E}[(Q-\theta)_+]\right\}.$$

For a receiver-agnostic rule $\varphi_0 = \varphi_0(F,\xi)$,

$$\mathbb{E}[q\varphi_0] = \mathbb{E}[\bar{q}\varphi_0], \qquad \bar{q} = \mathbb{E}[q \mid F,\xi].$$

The threshold identity therefore gives

$$j_0 = \min_{\theta\in[0,1]} \left\{\theta\rho_{\mathrm{long}} + \mathbb{E}[(\bar{q}-\theta)_+]\right\},$$

whereas receiver-aware routing gives

$$j_1 = \min_{\theta\in[0,1]} \left\{\theta\rho_{\mathrm{long}} + \mathbb{E}[(q-\theta)_+]\right\}.$$

For every θ, conditional Jensen's inequality gives

$$\mathbb{E}\left[(q-\theta)_+ \mid F, \xi\right] \geq (\bar{q}-\theta)_+.$$

Taking expectations and minimizing over θ therefore gives $j_1 \geq j_0$.

Let θ^* minimize the receiver-aware objective. If a positive-probability set of contexts satisfies

$$\Pr(q < \theta^* \mid F, \xi) > 0, \qquad \Pr(q > \theta^* \mid F, \xi) > 0,$$

then the conditional distribution crosses the hinge at θ^* and Jensen's inequality is strict on that set. Thus,

$$\mathbb{E}[(q-\theta^*)_+] > \mathbb{E}[(\bar{q}-\theta^*)_+].$$

Consequently,

$$j_1 > \theta^* \rho_{\text{long}} + \mathbb{E}[(\bar{q}-\theta^*)_+] \geq j_0,$$

which establishes the strict inequality. □

Illustrative separation. Consider $M \geq 2$ equally likely receiver states for a fixed context ξ. Suppose one receiver has utility $c \in (0,1)$ and the remaining receivers have zero utility. For $\rho_{\text{long}} = 1/M$, a receiver-agnostic rule must accept independently of receiver identity and obtains

$$j_0 = \frac{c}{M^2}.$$

A receiver-aware rule accepts exactly the positive-utility receiver state and obtains

$$j_1 = \frac{c}{M}.$$

Receiver conditioning therefore admits a factor-M separation in this construction.

C.3 Proof of Proposition 2

Using the block partition of $\mathbf{w}_F$ from the main text, define

$$b_F(C) = w_{F,0} + w_{F,\text{len}} \log(1+|C|) + w_{F,\kappa}\kappa_C + w_{F,\alpha}\alpha_C + (\mathbf{w}_{F,o} - \mathbf{w}_{F,\Delta})^\top \mathbf{s}_o,$$

together with

$$\mathbf{a}_F = \mathbf{w}_{F,r} + \mathbf{w}_{F,\Delta}, \qquad \mathbf{c}_F = \mathbf{w}_{F,I}.$$

For fixed F and origin state $\mathbf{s}_o$, the routing score becomes

$$s_F(C, \mathbf{s}_r) = b_F(C) + \mathbf{a}_F^\top \mathbf{s}_r + \kappa_C \mathbf{c}_F^\top \mathbf{s}_r.$$

Proof. For clauses C and D at receiver state $\mathbf{s}_r$,

$$s_F(C, \mathbf{s}_r) - s_F(D, \mathbf{s}_r) = b_F(C) - b_F(D) + (\kappa_C - \kappa_D)\mathbf{c}_F^\top \mathbf{s}_r.$$

Subtracting the corresponding difference at $\mathbf{s}_{r'}$ gives

$$[s_F(C, \mathbf{s}_r) - s_F(D, \mathbf{s}_r)] - [s_F(C, \mathbf{s}_{r'}) - s_F(D, \mathbf{s}_{r'})] = (\kappa_C - \kappa_D)\mathbf{c}_F^\top(\mathbf{s}_r - \mathbf{s}_{r'}).$$

Removing the interaction feature $\kappa_e \mathbf{s}_{r_e}$ sets $\mathbf{c}_F = \mathbf{0}$, making every pairwise clause ordering independent of the receiver state.

For the converse, assume $\kappa_{C_0} \neq \kappa_{C_1}$ and $\mathbf{s}_0 \neq \mathbf{s}_1$. Define

$$x(C) = \frac{\kappa_C - \kappa_{C_0}}{\kappa_{C_1} - \kappa_{C_0}}, \qquad t(\mathbf{s}) = \frac{(\mathbf{s}_1 - \mathbf{s}_0)^\top (\mathbf{s} - \mathbf{s}_0)}{\|\mathbf{s}_1 - \mathbf{s}_0\|_2^2}.$$

Then $x(C_0) = t(\mathbf{s}_0) = 0$ and $x(C_1) = t(\mathbf{s}_1) = 1$. The score

$$s(C, \mathbf{s}) = (2x(C) - 1)(2t(\mathbf{s}) - 1)$$

contains only a constant term, a term linear in κ_C, a term linear in $\mathbf{s}$, and a bilinear term $\kappa_C \mathbf{c}^\top \mathbf{s}$, all represented by the proposed scorer. The resulting preferences satisfy

$$s(C_0, \mathbf{s}_0) > s(C_1, \mathbf{s}_0), \qquad s(C_1, \mathbf{s}_1) > s(C_0, \mathbf{s}_1).$$

By contrast, any additive score $b(C) + h(\mathbf{s})$ satisfies

$$[b(C_0) + h(\mathbf{s})] - [b(C_1) + h(\mathbf{s})] = b(C_0) - b(C_1),$$

which cannot reverse across receiver states. □

C.4 Proof of Theorem 2

Proof. Both $\mathcal{S}_N^*$ and $\widehat{\mathcal{S}}_N$ contain the mandatory set $\mathcal{M}_N$, so these events cancel from the utility difference. Define

$$\mathcal{A} = \mathcal{S}_N^* \setminus \widehat{\mathcal{S}}_N, \qquad \mathcal{B} = \widehat{\mathcal{S}}_N \setminus \mathcal{S}_N^*.$$

The two sets have equal cardinality $h \le k_N$. Let $\varphi : \mathcal{A} \to \mathcal{B}$ be any bijection.

Because $\widehat{\mathcal{R}}_N$ contains the k_N highest-scoring non-mandatory events,

$$\widehat{p}_{\varphi(e)} \ge \widehat{p}_e \qquad \forall e \in \mathcal{A}.$$

By the definition of $\varepsilon_N^{\mathrm{rank}}$,

$$q_e - q_{\varphi(e)} \le \widehat{p}_e - \widehat{p}_{\varphi(e)} + \varepsilon_N^{\mathrm{rank}} \le \varepsilon_N^{\mathrm{rank}}.$$

Summing over $\mathcal{A}$ gives

$$\sum_{e \in \mathcal{S}_N^*} q_e - \sum_{e \in \widehat{\mathcal{S}}_N} q_e = \sum_{e \in \mathcal{A}} \left(q_e - q_{\varphi(e)}\right) \le h\varepsilon_N^{\mathrm{rank}} \le k_N \varepsilon_N^{\mathrm{rank}}.$$

□

C.5 Proof of Theorem 3 and Corollary 1

We first establish the dual interpretation of the deployed controller. Consider the fractional allocation problem

$$\max_{\{a_e\}} \sum_{e=0}^{N-1} s_e a_e \qquad \text{s.t.} \qquad a_e \in [m_e, 1], \quad \sum_{e=0}^{N-1} a_e = \rho N.$$

Its Lagrangian with multiplier θ is

$$\mathcal{L}(\mathbf{a}, \theta) = \rho N \theta + \sum_{e=0}^{N-1} (s_e - \theta) a_e.$$

Maximizing independently over $a_e \in [m_e, 1]$ gives

$$\sup_{\mathbf{a}} \mathcal{L}(\mathbf{a}, \theta) = \sum_{e=0}^{N-1} \psi_e(\theta),$$

where

$$\psi_e(\theta) = \rho\theta + m_e(s_e - \theta) + (1 - m_e)(s_e - \theta)_+.$$

At the deployed threshold decision,

$$g_e = \rho - a_e \in \partial\psi_e(\theta_e), \qquad |g_e| \le g_\rho = \max\{\rho, 1 - \rho\}.$$

Moreover,

$$\theta_{e+1} = \mathrm{proj}_{\mathcal{I}}\left(\theta_e - \eta g_e\right) = \mathrm{clip}_{\mathcal{I}}\left(\theta_e + \eta(a_e - \rho)\right),$$

so the deployed controller is projected subgradient descent.

Proof of Theorem 3. Convexity of ψ_e gives

$$\psi_e(\theta_e) - \psi_e(\vartheta_e) \le g_e(\theta_e - \vartheta_e).$$

Nonexpansiveness of Euclidean projection further gives

$$g_e(\theta_e - \vartheta_e) \le \frac{(\theta_e - \vartheta_e)^2 - (\theta_{e+1} - \vartheta_e)^2}{2\eta} + \frac{\eta g_e^2}{2}.$$

Summing over e and allowing the comparator to vary yields

$$\sum_{e=0}^{N-1} \left[(\theta_e - \vartheta_e)^2 - (\theta_{e+1} - \vartheta_e)^2\right] \le d_\theta^2 + 2d_\theta \upsilon_N.$$

Using $|g_e| \le g_\rho$ therefore gives

$$\sum_{e=0}^{N-1} [\psi_e(\theta_e) - \psi_e(\vartheta_e)] \le \frac{d_\theta^2 + 2d_\theta \upsilon_N}{2\eta} + \frac{\eta N g_\rho^2}{2}.$$

The argument is deterministic on the realized event stream and therefore requires neither independence nor stationarity. □

Proof of Corollary 1. When clipping is inactive,

$$\theta_{e+1} - \theta_e = \eta(a_e - \rho).$$

Summing over N routing decisions gives

$$\bar{a}_N - \rho = \frac{\theta_N - \theta_0}{\eta N}.$$

Since $\theta_0, \theta_N \in [\theta_{\min}, \theta_{\max}]$,

$$|\bar{a}_N - \rho| \le \frac{\theta_{\max} - \theta_{\min}}{\eta N}.$$

Every event in $\mathcal{M}_N$ is mandatory, so

$$\bar{a}_N \ge \frac{|\mathcal{M}_N|}{N}.$$

□

Connection to the communication constraint. For a solve of formula F with $N = n_F(\pi)$ routing events, $\bar{a}_N = \bar{a}_F(\pi)$. Hence, when clipping is inactive and the mandatory-transfer floor is feasible,

$$N \ge \frac{\theta_{\max} - \theta_{\min}}{\eta \varepsilon_\rho}$$

is sufficient for $|\bar{a}_F(\pi) - \rho| \le \varepsilon_\rho$, matching the communication-budget condition in Equation 2.

Tracking under active clipping. Define the projection residual

$$\zeta_e = \theta_{e+1} - \theta_e - \eta(a_e - \rho).$$

The clipped controller satisfies

$$\bar{a}_N - \rho = \frac{\theta_N - \theta_0 - \sum_{e=0}^{N-1} \zeta_e}{\eta N}.$$

Persistent threshold saturation therefore appears explicitly through the accumulated projection residual. Upper saturation corresponds to requested traffic below the attainable level, including the mandatory-transfer floor, whereas lower saturation indicates that the available score range cannot sustain the requested acceptance rate.

D Appendix: Experimental Details

D.1 Benchmark Details

The evaluation covers three regimes: in-distribution testing on the same broad problem families used for training, distribution-shift evaluation on unseen SATLIB families, and cross-problem transfer across distinct combinatorial encodings. All variable and clause counts are read directly from the DIMACS headers. Each CNF is evaluated over three repeated solver runs; repeats are not counted as additional benchmark instances.

In-distribution benchmarks. The in-distribution benchmark contains 1700 CNF formulas spanning random 3-SAT, graph coloring, and cryptographic instances. Random 3-SAT contributes 800 formulas across four scales, graph coloring contributes 600 formulas across three scales, and the cryptographic family contributes 300 formulas across three scales. The complete benchmark therefore produces 5100 solver runs per method.

Table 5: In-distribution benchmark composition and CNF scale

Family	Scale	CNFs	Variables	Clauses
3-SAT	250	200	250	1065
3-SAT	300	200	300	1278
3-SAT	350	200	350	1491
3-SAT	400	200	400	1704
Coloring	400	200	1200	4144–4603
Coloring	500	200	1500	5216–5750
Coloring	600	200	1800	6306–6861
Crypto	10	100	544–568	9947–11528
Crypto	15	100	537–552	9129–10321
Crypto	20	100	526–544	8182–9309

Random 3-SAT ranges from 250 to 400 variables and from 1065 to 1704 clauses, with a clause-to-variable ratio of approximately 4.26. Graph-coloring instances are larger, containing 1200–1800 variables and 4144–6861 clauses. Cryptographic instances contain 526–568 variables and substantially denser CNFs, with clause counts ranging from 8182 to 11528.

Table 6 summarizes the corresponding family-level statistics.

Table 6: Family-level statistics of the in-distribution benchmark

Family	CNFs	Variables (min/mean/max)	Clauses (min/mean/max)
3-SAT	800	250 / 325 / 400	1065 / 1384.5 / 1704
Coloring	600	1200 / 1500 / 1800	4144 / 5492.7 / 6861
Crypto	300	526 / 542.4 / 568	8182 / 9629.1 / 11528
Overall	1700	250 / 778.1 / 1800	1065 / 4289.4 / 11528

SATLIB distribution-shift benchmarks. The distribution-shift benchmark contains 100 retained CNFs from five SATLIB families not represented among the training families: AIM, Beijing, Dubois, Hanoi, and SSA. These formulas span substantial structural and scale variation, from 50 to 1931 variables and from 80 to 14468 clauses. Three repeated solver runs per formula produce 300 runs per method.

Table 7: SATLIB distribution-shift benchmark statistics

Family	CNFs	Variables (min/mean/max)	Clauses (min/mean/max)
AIM	72	50 / 116.7 / 200	80 / 322.8 / 1200
Beijing	5	125 / 464.8 / 708	310 / 1152.4 / 1702
Dubois	14	60 / 112.5 / 300	160 / 300.0 / 800
Hanoi	2	718 / 1324.5 / 1931	4934 / 9701.0 / 14468
SSA	7	435 / 1199.7 / 1501	1027 / 2775.7 / 3575
Overall	100	50 / 233.5 / 1931	80 / 720.3 / 14468

AIM contains the largest number of formulas and primarily covers small-to-medium instances, whereas Hanoi contains only two formulas but reaches the largest scale in both variables and clauses. SSA further extends evaluation to larger structured formulas, while Beijing and Dubois provide intermediate structural regimes.

The original SATLIB evaluation manifest contains 113 formulas. Pigeon-hole and Pret instances are excluded consistently from the reported subset for every compared method, leaving the 100 formulas summarized above.

Cross-problem transfer benchmarks. The cross-problem benchmark contains 600 CNFs derived from Dominating Set, Clique, and Coloring encodings following Interian & Bernardini (2023). Each problem contributes 200 formulas and is evaluated separately from the in-distribution and SATLIB benchmarks. The resulting benchmark measures transfer across distinct combinatorial encodings while retaining a common CNF solver interface.

Table 8: Cross-problem transfer benchmark statistics

Problem	CNFs	Variables (min/mean/max)	Clauses (min/mean/max)
Dominating Set	200	60 / 60 / 60	730 / 731.8 / 732
Clique	200	60 / 60 / 60	1716 / 1736.9 / 1758
Coloring	200	100 / 100 / 100	600 / 665.1 / 730
Overall	600	60 / 73.3 / 100	600 / 1044.6 / 1758

Dominating Set instances contain 60 variables and approximately 732 clauses. Clique uses the same number of variables but substantially denser encodings, with approximately 1737 clauses on average. Coloring contains 100 variables and approximately 665 clauses on average. These differences expose the router to distinct clause-density and encoding structures beyond the SAT families used for training.

D.2 Baseline Details

Classical CDCL solvers. Kissat Biere et al. (2022) and Glucose Audemard & Simon (2009) serve as single-worker CDCL references. Kissat runs in a fresh native process for each instance, while Glucose uses the archived configuration integrated into the evaluation harness.

Parallel CDCL. Glucose-Syrup Audemard & Simon (2017) provides the native parallel backend underlying FOCON. Both solvers use eight workers in the main in-distribution evaluation. Glucose-Syrup retains its native learned-clause sharing policy, whereas FOCON modifies only clause communication. Branching, propagation, conflict analysis, and the remaining CDCL search procedure are unchanged, enabling a direct comparison of clause-routing policies under the same parallel solver backend.

Learning-guided solvers. RLAF Tönshoff & Grohe (2026) introduces learned guidance into SAT search and uses its saved checkpoint under the archived single-instance evaluation protocol. NeuroBack Wang et al. (2024) augments Kissat with neural phase prediction and runs each instance in a fresh process. Its reported end-to-end runtime includes model loading, graph construction, neural inference, and subsequent native solving.

Local search. WalkSAT Tönshoff & Grohe (2026) serves as a stochastic local-search reference, using noise 0.5 and a 60s timeout. Since WalkSAT does not certify unsatisfiability, its all-instance results are reported alongside the complete CDCL-based solvers rather than interpreted as a direct replacement for complete SAT solving.

D.3 Implementation and Evaluation Protocol

Table 9 summarizes the implementation, training, and evaluation settings used throughout the experiments.

Table 9: FOCON implementation and evaluation settings

Parameter	Setting
Hidden channels	64
Message-passing steps	3
Formula embedding dimension	32
Training epochs	50
Batch size	4
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	10^{-4}
Gradient clipping	1.0
Event cap per formula	2048
Target acceptance rate	0.20
Controller step size	0.002
OOD calibration quantile	0.99
Model selection criterion	Worst-domain validation F1
Parallel backend	Glucose-Syrup, 8 workers
Timeout	60s

E Appendix: Additional Experiments

This appendix expands the aggregate results reported in Tables 1 and 2, and provides diagnostic analyses of support-aware fallback, paired solve outcomes, and solve-time behavior. All reported standard deviations are computed across the three repeated solver evaluations of the fixed checkpoint.

E.1 Detailed In-Distribution Results

Table 10 expands the family-level results in Table 1 with solved-trial counts. Each 3-SAT instance is evaluated three times for 2400 trials, Coloring contains 1800 trials, and Crypto contains 900 trials.

Table 10: Detailed in-distribution performance by benchmark family

Family	Solver	Solved / trials	SR (%) ↑	RT (s) ↓	PAR-2 (s) ↓
3-SAT	Kissat	1832 / 2400	$\mathbf{76.33 \pm 0.07}$	$\mathbf{17.22 \pm 0.07}$	$\mathbf{31.41 \pm 0.11}$
	Glucose	1417 / 2400	59.04 ± 0.07	29.62 ± 0.01	54.00 ± 0.04
	Glucose-Syrup	1726 / 2400	71.92 ± 0.44	22.75 ± 0.38	39.58 ± 0.64
	WalkSAT	1099 / 2400	45.79 ± 0.31	35.44 ± 0.29	67.96 ± 0.47
	NeuroBack	1831 / 2400	76.29 ± 0.07	19.65 ± 0.07	33.84 ± 0.08
	RLAF	1767 / 2400	73.63 ± 0.00	22.89 ± 0.00	38.17 ± 0.00
	FOCON	1753 / 2400	73.04 ± 0.31	21.71 ± 0.14	37.86 ± 0.32
Coloring	Kissat	1657 / 1800	92.06 ± 0.10	9.21 ± 0.10	13.97 ± 0.09
	Glucose	1569 / 1800	87.17 ± 0.00	12.38 ± 0.01	20.03 ± 0.00
	Glucose-Syrup	1698 / 1800	94.33 ± 0.44	7.35 ± 0.19	10.74 ± 0.45
	WalkSAT	0 / 1800	0.00 ± 0.00	60.00 ± 0.00	120.00 ± 0.00
	NeuroBack	1655 / 1800	91.94 ± 0.42	12.49 ± 0.16	17.31 ± 0.40
	RLAF	1652 / 1800	91.78 ± 0.10	10.18 ± 0.00	14.94 ± 0.06
	FOCON	1701 / 1800	$\mathbf{94.50 \pm 0.29}$	$\mathbf{6.90 \pm 0.05}$	$\mathbf{10.19 \pm 0.22}$
Crypto	Kissat	701 / 900	77.89 ± 0.19	18.64 ± 0.00	31.89 ± 0.12
	Glucose	723 / 900	80.33 ± 0.00	18.88 ± 0.01	30.54 ± 0.00
	Glucose-Syrup	878 / 900	97.56 ± 0.84	9.60 ± 0.25	11.06 ± 0.73
	WalkSAT	0 / 900	0.00 ± 0.00	60.00 ± 0.00	120.00 ± 0.00
	NeuroBack	673 / 900	74.78 ± 0.19	25.02 ± 0.04	40.11 ± 0.10
	RLAF	852 / 900	94.67 ± 0.00	13.86 ± 0.01	16.94 ± 0.01
	FOCON	884 / 900	$\mathbf{98.22 \pm 0.51}$	$\mathbf{9.08 \pm 0.23}$	$\mathbf{10.14 \pm 0.53}$

The decomposition sharpens the family-level pattern in the main results. FOCON improves over the matched Glucose-Syrup backend on all three families, but the magnitude depends strongly on formula structure. On 3-SAT, the gain over native sharing is modest and Kissat remains more effective. On Coloring and Crypto, learned routing improves both solve coverage and runtime, with the largest PAR-2 reduction occurring on Crypto. The result therefore does not reflect a uniform

advantage across SAT distributions; the benefit is more pronounced on the structured families where formula context and receiver-specific search state provide richer routing signals.

E.2 Detailed Generalization Results

Table 11 expands the OOD portion of Table 2 by reporting solve coverage and runtime in addition to PAR-2.

Table 11: Detailed performance across retained unseen SATLIB families

Family	Solver	Solved / trials	SR (%) ↑	RT (s) ↓	PAR-2 (s) ↓
Beijing	NeuroBack	15 / 15	100.00 ± 0.00	8.085 ± 0.240	8.085 ± 0.240
	RLAF	15 / 15	100.00 ± 0.00	1.457 ± 0.340	1.457 ± 0.340
	FOCON	15 / 15	100.00 ± 0.00	$\mathbf{0.495 \pm 0.071}$	$\mathbf{0.495 \pm 0.071}$
AIM	NeuroBack	216 / 216	100.00 ± 0.00	7.127 ± 0.006	7.127 ± 0.006
	RLAF	216 / 216	100.00 ± 0.00	$\mathbf{0.021 \pm 0.000}$	$\mathbf{0.021 \pm 0.000}$
	FOCON	216 / 216	100.00 ± 0.00	0.021 ± 0.003	0.021 ± 0.003
Dubois	NeuroBack	39 / 42	92.86 ± 0.00	6.594 ± 0.064	15.165 ± 0.064
	RLAF	39 / 42	92.86 ± 0.00	$\mathbf{0.021 \pm 0.001}$	$\mathbf{8.591 \pm 0.001}$
	FOCON	39 / 42	92.86 ± 0.00	0.022 ± 0.004	8.592 ± 0.004
Hanoi	NeuroBack	6 / 6	100.00 ± 0.00	7.665 ± 0.092	7.665 ± 0.092
	RLAF	6 / 6	100.00 ± 0.00	1.312 ± 0.007	1.312 ± 0.007
	FOCON	6 / 6	100.00 ± 0.00	$\mathbf{0.068 \pm 0.018}$	$\mathbf{0.068 \pm 0.018}$
SSA	NeuroBack	21 / 21	100.00 ± 0.00	7.854 ± 0.067	7.854 ± 0.067
	RLAF	21 / 21	100.00 ± 0.00	$\mathbf{0.028 \pm 0.001}$	$\mathbf{0.028 \pm 0.001}$
	FOCON	21 / 21	100.00 ± 0.00	0.030 ± 0.011	0.030 ± 0.011

Except for Dubois, all three methods solve every retained SATLIB instance. Consequently, the differences on Beijing, AIM, Hanoi, and SSA primarily reflect solving time rather than coverage. Dubois exhibits the same 92.86% SR for all three methods, causing unsolved trials to dominate PAR-2 despite very short runtime on solved instances. The large reductions on Beijing and Hanoi therefore arise from faster completed solves, whereas the near-equal AIM, Dubois, and SSA results indicate families for which routing provides little additional benefit over RLAF.

Table 12 reports the corresponding cross-problem results. Since every evaluated instance is solved, RT and PAR-2 coincide.

Table 12: Detailed performance across combinatorial SAT encodings

Problem	Solver	Solved / trials	SR (%) ↑	RT (s) ↓	PAR-2 (s) ↓
Dominating Set	NeuroBack	600 / 600	100.00 ± 0.00	2.860 ± 0.009	2.860 ± 0.009
	RLAF	600 / 600	100.00 ± 0.00	3.129 ± 0.006	3.129 ± 0.006
	FOCON	600 / 600	100.00 ± 0.00	$\mathbf{2.801 \pm 0.007}$	$\mathbf{2.801 \pm 0.007}$
Clique	NeuroBack	600 / 600	100.00 ± 0.00	2.714 ± 0.004	2.714 ± 0.004
	RLAF	600 / 600	100.00 ± 0.00	2.956 ± 0.002	2.956 ± 0.002
	FOCON	600 / 600	100.00 ± 0.00	$\mathbf{2.655 \pm 0.002}$	$\mathbf{2.655 \pm 0.002}$
Coloring	NeuroBack	600 / 600	100.00 ± 0.00	3.461 ± 0.009	3.461 ± 0.009
	RLAF	600 / 600	100.00 ± 0.00	3.868 ± 0.007	3.868 ± 0.007
	FOCON	600 / 600	100.00 ± 0.00	$\mathbf{3.397 \pm 0.006}$	$\mathbf{3.397 \pm 0.006}$

Because solve coverage saturates at 100%, the cross-problem experiment isolates execution efficiency among the three learning-guided methods compared in the main generalization table. FOCON remains faster on all three encodings, but the margins are substantially smaller than on Beijing and Hanoi, indicating transfer of the learned routing rule without requiring a large distribution-specific gain.

E.3 Adaptive Communication-Budget Control

Table 13 evaluates whether online threshold adaptation can enforce different communication budgets beyond the default operating point. The fixed threshold remains confined to a narrow acceptance

Table 13: Adaptive communication-budget control on the in-distribution benchmark

	Accept. (%)		SR (%) ↑		PAR-2 (s) ↓	
Target	Fixed	Adaptive	Fixed	Adaptive	Fixed	Adaptive
5%	11.27	**5.01**	77.76	**83.47**	32.11	**25.43**
10%	9.38	**9.98**	77.18	**79.76**	32.89	**29.62**
20%	9.51	**19.92**	77.41	**79.29**	32.74	**30.00**
40%	9.81	**39.89**	77.18	**82.65**	32.91	**26.55**

range of 9.38–11.27% as the target varies, whereas adaptive control closely tracks all requested rates from 5% to 40%. This regulation consistently improves SR and PAR-2 over the corresponding fixed-threshold variant. Performance is not monotonic in the amount of accepted traffic, indicating that effective clause communication depends on maintaining a controlled routing regime rather than simply increasing transfer volume.

E.4 Support-Aware OOD Routing

Table 14: Support-aware routing behavior across retained OOD families

Family	OOD score	Abstention (%)
Beijing	0.32	0.00
SSA	0.37	0.00
Hanoi	0.56	0.00
AIM	1.22	33.33
Dubois	2.08	92.86

Table 14 shows a clear separation in deployment behavior across unseen families. Learned routing remains active throughout Beijing, SSA, and Hanoi, becomes selective on AIM, and falls back to native sharing for most Dubois instances. The gate therefore does not apply a uniform OOD response: formulas close to the calibrated support retain learned routing, whereas structurally distant families increasingly defer to the native policy. This behavior complements the component ablation in Table 4, where removing support-aware fallback degrades OOD SR and PAR-2.

E.5 Paired Solve Outcomes

Table 15: Paired solve outcomes on matched in-distribution evaluation trials

Baseline	FOCON only	Baseline only	Both	Neither
Glucose-Syrup	**86**	50	4252	712
RLAF	**156**	89	4182	673
Kissat	**359**	211	3979	551
NeuroBack	**391**	212	3947	550

Aggregate SR does not show whether two solvers fail on the same trials. Table 15 therefore joins outcomes by formula and repeated run. Against the matched Glucose-Syrup backend, FOCON uniquely solves 86 trials while Glucose-Syrup uniquely solves 50. The asymmetry is larger against RLAF, Kissat, and NeuroBack. Hence, the aggregate SR gain does not arise solely from identical solved sets with small timing changes; learned routing also changes which search trajectories reach a solution within the timeout. These counts are descriptive paired outcomes and do not constitute a significance test.

E.6 Solve-Time Profile

To isolate the effect of learned clause routing under the matched parallel backend, Table 16 compares the fraction of all in-distribution trials solved before several runtime deadlines by FOCON and Glucose-Syrup. Unsolved trials remain in the denominator.

Table 16: Solved-by-deadline rates for FOCON and the matched Glucose-Syrup backend

Solver	0.1 s	1 s	5 s	10 s	30 s	60 s
Glucose-Syrup	**14.67**	39.53	60.84	68.00	78.04	84.35
FOCON	10.61	**40.10**	**62.88**	**69.61**	**79.55**	**85.06**

Native sharing solves a larger fraction of extremely short cases within 0.1s, whereas FOCON overtakes Glucose-Syrup by the 1s deadline and maintains the advantage thereafter. The gap grows from 0.57 at 1s to 2.04 at 5s and remains positive through the timeout horizon. This profile suggests that learned routing contributes primarily after trivial solves have already terminated, where selective clause exchange has sufficient search history to influence subsequent propagation and conflict reuse.