# CONSISTENCY-DRIVEN NEURO-SYMBOLIC SAT ENSEMBLE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Neural networks have recently emerged as a third paradigm for Boolean satisfiability (SAT) solving, alongside complete and stochastic local-search solvers, with independently trained models offering complementary predictions on the same formula, yet frequently disagreeing on the assignments they produce. To date, no ensemble method has been developed for aggregating the outputs of multiple neural SAT solvers, and a naive application of classical ensemble techniques would only reduce this disagreement to scalar per-variable metrics such as vote counts, discarding the relational structure between variables that is essential for constructing a consistent satisfying assignment. We present CDNSE, a new consistency-driven ensemble method for neuro-symbolic SAT solvers. We develop a mathematical framework to characterize consistency among solver outputs, capturing both agreement on variable assignments and on relationships between variables. These equivalence relations allow us to group variables where the ensemble shows consistent voting, which significantly reduces the search space for local search. We benchmark CDNSE against eleven fusion-based and ten learning-based methods on 123 instances from six SATLIB datasets. cdNSE improves the solve rate by a relative improvement of $14.1\%$ over the strongest baseline and requires $27.1\%$ fewer flips to find a solution. When we increase the flip-budget for the local search engine, cdNSE achieves a $91.1\%$ solve rate, compared to $83.7\%$ for the strongest baseline method.

## 1 INTRODUCTION

Boolean satisfiability (SAT) is the canonical NP-complete problem (Cook, 1971) and underpins a broad range of applications. Existing SAT solvers can be broadly divided into two dominant paradigms. The first comprises complete solvers, most notably conflict-driven clause learning (CDCL) methods, which derive their effectiveness from proof-theoretic mechanisms including clause learning, non-chronological backtracking, and activity-based branching heuristics (Marques-Silva & Sakallah, 1999; Moskewicz et al., 2001; Eén & Sörensson, 2004; Marques-Silva et al., 2021). The second comprises incomplete stochastic local-search methods, which formulate SAT solving as the optimisation of an energy-like objective and iteratively navigate the resulting search landscape (Selman et al., 1994; Thornton et al., 2004; Cai & Su, 2013).

A third paradigm has emerged with the application of neural networks to SAT solving. These methods encode a Boolean formula as a structured representation, typically a graph, and learn to predict satisfying assignments or related solution signals (Selsam et al., 2019; Amizadeh et al., 2019a; Ozoliņš et al., 2022; Cameron et al., 2020). Unlike traditional solvers, neural SAT solvers can exploit learned representations of formula structure and provide complementary predictions from independently trained models. However, existing neural SAT solvers are generally trained and evaluated as individual models, often using different architectures, graph encodings, and training procedures. Their performance can also degrade substantially on out-of-distribution instances, where the learned representations may not generalise to unseen formula distributions. More importantly, different neural solvers frequently produce inconsistent assignments for the same instance, creating a fundamental challenge for combining their predictions effectively.

Ensemble methods can be a promising solution for these above problems. In particular, this method provides a natural framework for exploiting multiple predictive models and has a long history of successful aggregation strategies, including bagging (Breiman, 1996), random subspaces (Ho, 1998), boosting (Freund & Schapire, 1997; Friedman, 2001), Bayesian model averaging, diversity-



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.

aware weighting (Liu & Yao, 1999; Kuncheva & Whitaker, 2003), and stacked generalisation (Wolpert, 1992). These approaches, however, are primarily designed to aggregate predictions by exploiting individual model reliability or diversity. Their effectiveness is commonly associated with the assumption that ensemble members make sufficiently uncorrelated errors, such that combining their predictions can mitigate individual model failures (Krogh & Vedelsby, 1994; Dietterich, 2000). This assumption is particularly restrictive for neural SAT solvers: their predictions are structured assignments over the same set of Boolean variables, and disagreement between models may itself exhibit strong dependencies.

As we develop ensemble strategies for neural SAT solving, we anticipate that the primary challenge will lie not merely in quantifying disagreement, but in representing its underlying structure. Specifically, aggregating multiple solvers' assignments typically reduces disagreement to scalar, per-variable metrics such as vote counts, confidence scores, or predictive entropy. However, such representations discard inter-variable relationships that are crucial for constructing a consistent satisfying assignment. For instance, scalar metrics cannot distinguish between two variables that exhibit perfectly correlated disagreement (where any model that flips the first variable also flips the second) and two variables that disagree entirely independently. By ignoring these structural dependencies, scalar representations lose vital information needed to guide the search for a valid solution. These two patterns are indistinguishable under independent per-variable aggregation, despite potentially requiring fundamentally different corrections during subsequent search. Therefore, the primary gap is not the absence of an additional scalar measure of ensemble confidence, but rather the lack of an explicit representation of the relational structure of agreement and disagreement among satisfying assignments produced by solvers.

To the best of our knowledge, no existing work has investigated ensemble methods that aggregate the outputs of multiple neural SAT solvers. The closest lines of research concern neural solver selection (Gao et al., 2024; Zhang et al., 2024; Malitsky et al., 2012) and multi-solver benchmarking frameworks (Li et al., 2024; Mojžíšek et al., 2025). These studies consider either selecting an appropriate solver for a given instance or evaluating multiple solvers independently, rather than jointly exploiting their predicted assignments to construct a solution. To address this gap, we introduce a **c**onsistency-**d**riven **N**euro-**S**ymbolic SAT **E**nsemble (CDNSE). Our contributions are summarised as follows:

1. We formulate agreement among solver outputs as an algebraic object that captures both agreement on variable values and agreement on pairwise relations between variables (§3.2). These two levels of agreement induce a partition of variables into subgroups, or blocks, within which the ensemble exhibits consistent voting behaviour.
2. We integrate the two levels of agreement into an existing local-search engine (§3.3). CDNSE (Algorithm 1) uses the additive clause weighting of Thornton et al. (2004), preceded by an unweighted WALKSAT/SKC warm-up phase (Selman et al., 1994), and performs block flipping induced by the proposed relations. This structure substantially reduces the search space of unrestricted $k$-out-of-$n$ random flips.
3. We evaluate CDNSE against eleven fusion-based and ten learning-based competitors on 123 instances spanning six SATLIB datasets (§4). CDNSE achieves a solve rate of $69.5\%$, compared with $60.9\%$ for the strongest baseline, corresponding to an absolute improvement of $8.6$ percentage points, or a relative improvement of $14.1\%$. It also requires fewer local-search flips, with a median of $474$ compared with $650$ for the strongest baseline. Moreover, as the flip budget increases, the advantage of CDNSE becomes more pronounced: under the larger flip budget, CDNSE achieves a solve rate of $91.1\%$, compared with $83.9\%$ for the strongest competitor.

## 2 RELATED WORKS

**Neural-network based SAT solvers.** The modern line of learning-based SAT solving begins with NEUROSAT (Selsam et al., 2019), which trains a message-passing network on the literal–clause incidence graph using only binary satisfiability supervision and recovers assignments by clustering the resulting literal embeddings. Subsequent approaches have varied primarily along three dimensions: graph representation, supervision, and the form of the network output. Amizadeh et al. (2019a), for example, replace the CNF graph with a circuit representation and train a differentiable relaxation

without explicit labels. Their PDP framework (Amizadeh et al., 2019b) further interprets message passing as a propagation–decimation procedure and generalises this paradigm to a broader family of constraint-satisfaction solvers within a unified architecture. QUERYSAT (Ozoliņš et al., 2022) introduces an explicit query mechanism that allows the network to evaluate candidate assignments against the formula during inference, rather than relying solely on a final prediction. Other approaches target the decision problem more directly: Cameron et al. (2020) develop an architecture that explicitly accounts for permutation invariance over clauses, while Prates et al. (2019) demonstrate that related neural formulations can be transferred to other NP-complete decision problems.

**Model ensembles.** Assembling is the conventional response to having access to $M$ predictive models. Classical ensemble methods differ primarily in how they induce diversity among their members. Bagging achieves diversity through bootstrap resampling of the training data (Breiman, 1996), while random subspace methods resample the feature space (Ho, 1998); random forests combine both forms of randomisation (Breiman, 2001). Boosting instead sequentially reweights training examples, encouraging successive members to specialise on previously difficult cases (Freund & Schapire, 1997; Friedman, 2001). Negative-correlation learning explicitly penalises agreement among ensemble members (Liu & Yao, 1999), whereas error-correcting output codes distribute a multi-class decision across a set of redundant binary prediction problems (Dietterich & Bakiri, 1995). Stacked generalisation learns the combination rule from held-out predictions (Wolpert, 1992), while Bayesian model averaging assigns model weights according to their posterior probabilities (Hoeting et al., 1999). A comprehensive account of the principles underlying these approaches is given by Dietterich (2000).

# 3 THE PROPOSED METHODS

## 3.1 PRELIMINARIES

Let $\mathcal{F}$ be a CNF formula over variables $\mathcal{V} = \{1, \dots, n\}$ with clauses $C_1, \dots, C_m$. An assignment is a vector $\mathbf{x} \in \{0,1\}^n$. The energy of $\mathbf{x}$ is defined as the number of unsatisfied clauses,

$$E(\mathbf{x}) = \big|\{\, j \in \{1, \dots, m\} : \mathbf{x} \not\models C_j \,\}\big|, \qquad E(\mathbf{x}) = 0 \iff \mathbf{x} \models \mathcal{F}. \tag{1}$$

Write $d_{\mathrm{H}}$ for Hamming distance and $|\mathbf{x}| = \sum_v x_v$ for Hamming weight. The search space is the Boolean lattice $B_n = (\{0,1\}^n, \subseteq)$, graded by Hamming weight (Figure 1). Edges join assignments differing in one variable ($d_{\mathrm{H}} = 1$); i.e., a single variable flip is exactly one lattice edge.

Given $M$ neural solvers/agents (throughout, we refer a neural solver as an agent, to distinguish with "model" $\mathbf{x} \models \mathcal{F}$)), the collection of agents are $\mathcal{A} = \{1, \dots, M\}$. Agent $i$ contributes its best recorded assignment $\mathbf{x}_i \in \{0,1\}^n$, that is, the lowest-energy assignment in Equation (1). We collect $M$-agent's assignments and form the *ensemble matrix* $\mathbf{X} \in \{0,1\}^{M \times n}$ with $\mathbf{X}_{i,v} = \mathbf{x}_i[v]$, and write $E_i = E(\mathbf{x}_i)$.

*Agent reliability* is encoded by a weight vector $\mathbf{w}$,

$$\mathbf{w}_i = \frac{\exp\big(-(E_i - E_{\min})/T\big)}{\sum_{k=1}^{M} \exp\big(-(E_k - E_{\min})/T\big)}, \qquad T = \max(\sigma_E,\, 1), \tag{2}$$

where $E_{\min} = \min_k E_k$ and $\sigma_E$ is the standard deviation of $\{E_k\}_{k=1}^{M}$ on the instance. $T$ is set per instance rather than tuned: energy gaps are measured in units of the ensemble's spread, and the one-clause floor prevents near-hard selection when agents differ by a single clause. Since $M\sigma_E^2 \geq (E_{\max} - E_{\min})^2/2$, $\max_i \mathbf{w}_i / \min_i \mathbf{w}_i \leq e^{\sqrt{2M}}$, so the weight values do not degenerates to a one-hot selector. We subtracting the minimum energy to prevent overflow/underflow in the softmax function, while the sum of the exponentials in the denominator performs the actual normalization so the weights sum to 1.

## 3.2 TWO-LEVEL ALGEBRA OF CONSISTENCY

### 3.2.1 AGREEMENT ON VALUES

For a variable $v \in \mathcal{V}$, we define the agreement relation $\sim_v$ in the agent set $\mathcal{A}$ by $i \sim_v j \iff \mathbf{x}_i[v] = \mathbf{x}_j[v]$; i.e., two agents $i$ and $j$ agree on the value of variable $v$. The defined relation $\sim_v$ is

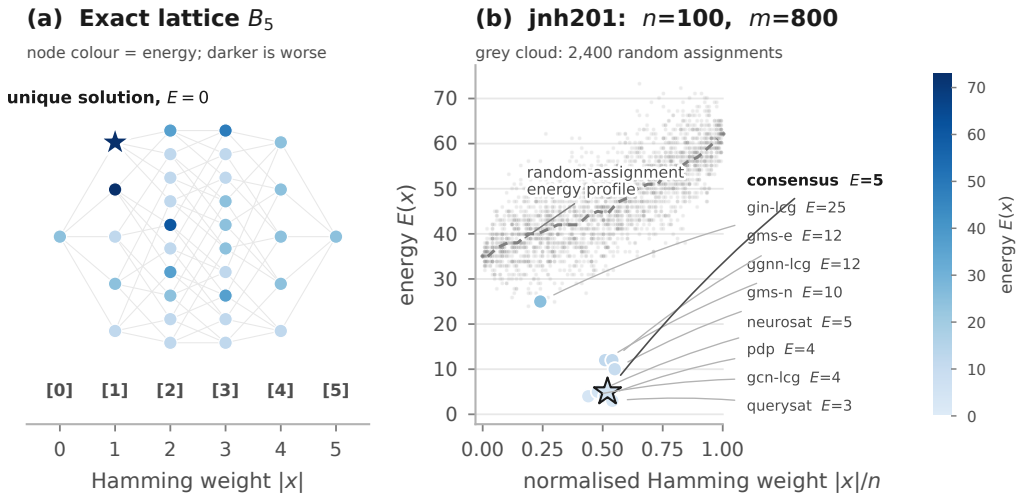



Figure 1: **The search space. (a)** The lattice $B_5$ of a 5-variable formula with a unique solution. **(b)** The lattice corresponding to a 100-variables, 800-clause instance.

an equivalence relation of $\mathcal{A}$ with at most two classes (or possibilities), including $\{i : \mathbf{x}_i[v] = 0\}$ or $\{i : \mathbf{x}_i[v] = 1\}$ since $\mathbf{x}$ is a Boolean vector. We define a *consistency coefficient* of $v$ as the weight mass of its dominant class,

$$\kappa_w(v) = \max_{b\in\{0,1\}} \sum_{i:\mathbf{x}_i[v]=b} \mathbf{w}_i, \quad c(v) = \arg\max_{b\in\{0,1\}} \sum_{i:\mathbf{x}_i[v]=b} \mathbf{w}_i. \tag{3}$$

Normalisation in Equation (2) makes $\kappa_w$ of Equation (3) a probability. Besides, $\kappa_w(v) \in [\frac{1}{2}, 1]$ for every $v \in \mathcal{V}$, with $\kappa_w(v) = 1$ iff the agents are unanimous[1] on $v$. This is because the two classes of $\sim_v$ partition $\mathcal{A}$, so their weight masses $s_0 = \sum_{i:\mathbf{x}_i[v]=0} \mathbf{w}_i$ and $s_1 = 1 - s_0$ sum to 1 by Equation (2). Then, $\kappa_w(v) = \max(s_0, s_1) \geq \frac{1}{2}$, since the larger of two numbers summing to 1 is at least $\frac{1}{2}$, and $\kappa_w(v) \leq 1$ since $s_0, s_1 \geq 0$. Equality at 1 forces the minority mass to vanish, and every weight in Equation (2) is strictly positive, so the minority class is empty.

The $\tau$-*core* is defined as $C_\tau = \{v : \kappa_w(v) \geq \tau\}$; i.e., the variables on which agreement is at least $\tau$. The *contested set* is its complement $D_\tau = \mathcal{V} \setminus C_\tau$. A special case is $C_1$, namely the *unanimous core*. Besides, $\tau = \frac{1}{2}$ (or $C_{\frac{1}{2}}$) denotes a balanced split, where half of the agents vote for one value and the remaining half vote for the other.

### 3.2.2 Agreement of Relations

The introduced value agreement in §3.2.1 is not necessarily beneficial. For example, if all agents unanimously vote for the same value $v$, but this value is incorrect, enforcing this agreement can be detrimental to the ensemble. We observe that two agents may disagree about the value of $u$ and $v$ while agreeing about how $u$ and $v$ relate. Formalizing this concept requires a two-step protocol: first, a single agent proposes a phase, and subsequently, the ensemble reaches a consensus.

For an agent $i \in \mathcal{A}$ and variables $u, v \in \mathcal{V}$, write

$$\delta_i(u, v) \;=\; \mathbf{x}_i[u] \oplus \mathbf{x}_i[v] \;\in\; \{0, 1\}, \qquad a \oplus b = a + b \pmod 2, \tag{4}$$

the *phase* that agent $i$ assigns to the pair: $\delta_i(u, v) = 0$ when $i$ gives $u$ and $v$ the same value, and $\delta_i(u, v) = 1$ when it separates them. Each agent alone induces an equivalence relation on $\mathcal{V}$, namely $u \sim^{(i)} v \iff \delta_i(u, v) = 0$, whose two classes are that agent's 0-set and 1-set.

Collecting the per-agent phases into the *phase vector* gives

$$\boldsymbol{\delta}(u, v) \;=\; \big(\delta_1(u, v), \ldots, \delta_M(u, v)\big) \;=\; \mathbf{X}_{:,u} \oplus \mathbf{X}_{:,v} \;\in\; \{0, 1\}^M, \tag{5}$$

[1] A unanimous decision occurs when all agents involved in a vote.

the ensemble-level relation requires the $M$ agents to agree on the phase, whatever it is:

$$u \sim_r v \iff \delta_i(u,v) = \delta_j(u,v)\ \forall i,j \in \mathcal{A} \iff \boldsymbol{\delta}(u,v) \in \{\mathbf{0}_M, \mathbf{1}_M\}. \tag{6}$$

The two admissible vectors are the two ways the ensemble can be unanimous about the pair: $\boldsymbol{\delta} = \mathbf{0}_M$ says every agent gives $u$ and $v$ the same value, $\boldsymbol{\delta} = \mathbf{1}_M$ says every agent separates them.

**Theorem 1.** *$\sim_r$ is an equivalence relation on $\mathcal{V}$.*

### 3.2.3 Agreement-informed block structure of ensemble matrix

One can view agreement on values $\sim_v$ in §3.2.1 as stratification of variables by agents. In contrast, the transitivity properties of agreement on relations $\sim_r$ in §3.2.2 is variable partition within an agent. The grouping-by-$\sim_v$ is performed by matching the consistency coefficient $\kappa(v)$ defined in Equation (3). For grouping-by-$\sim_r$, we define the *column code* and its complement as

$$z(v) = \sum_{i=1}^{M} \mathbf{x}_i[v]\, 2^{i-1} \in \{0, \ldots, 2^M - 1\}, \qquad \bar{z}(v) = (2^M - 1) - z(v), \tag{7}$$

so $z(v)$ is the column of $\mathbf{X}$ at $v$ read as an $M$-bit integer and $\bar{z}(v)$ is its bitwise complement. The variable partition by relation agreement $\sim_r$ is by matching the canonical code

$$\zeta(v) = \min(z, \bar{z}). \tag{8}$$

For instance, we construct the ensemble matrix $\mathbf{X}_{8\times 14}$ from 8 agents on a SAT instance with 14 variables as in Table 1.

Table 1: Illustrative example of agreement-informed block structure in ensemble matrix

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ | $v_{11}$ | $v_{12}$ | $v_{13}$ | $v_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Agent 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Agent 2 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Agent 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Agent 4 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Agent 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Agent 6 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Agent 7 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Agent 8 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| $\kappa(v)$ | **7/8** | 5/8 | **7/8** | **7/8** | 1 | **7/8** | 6/8 | 7/8 | 1 | 1 | 6/8 | 7/8 | 7/8 | 7/8 |
| $z(v)$ | 64 | 81 | 223 | 32 | 0 | 64 | 95 | 223 | 0 | 0 | 95 | 223 | 32 | 223 |
| $\bar{z}(v)$ | 191 | 174 | 32 | 223 | 255 | 191 | 160 | 32 | 255 | 255 | 160 | 32 | 223 | 32 |
| $\zeta(v)$ | 64 | 81 | 32 | 32 | 0 | 64 | 95 | 32 | 0 | 0 | 95 | 32 | 32 | 32 |
| Block | $B_1$ | $B_2$ | $B_3$ | $B_3$ | $B_4$ | $B_1$ | $B_5$ | $B_3$ | $B_4$ | $B_4$ | $B_5$ | $B_3$ | $B_3$ | $B_3$ |

Here, grouping $v_3$ and $v_4$ under $\sim_v$ with

$$\begin{aligned}\kappa(v_3) &= \max_{b\in\{0,1\}} \sum_{i:\mathbf{x}_i[v_3]=b} \mathbf{w}_i = \max\Big(\underbrace{\tfrac{1}{8}}_{\text{agent 3}}, \underbrace{\tfrac{1}{8}+\tfrac{1}{8}+\tfrac{1}{8}+\tfrac{1}{8}+\tfrac{1}{8}+\tfrac{1}{8}+\tfrac{1}{8}}_{\text{agents 1,2,4,5,6,7,8}}\Big) \\ &= \max\big(\tfrac{1}{8}, \tfrac{7}{8}\big) = \tfrac{7}{8},\end{aligned} \tag{9}$$

and $c(v_3) = 1$. This indicates that "Agent 3 disagrees with the remaining seven agents". However, $\sim_v$ alone does not identify "*which agent is in disagreement*". Both $v_1$ and $v_3$ (as well as $v_4$) exhibit a 1-against-7 consensus pattern, yet the minority agent differs: Agent 2 is the minority in $v_1$, whereas Agent 3 is the minority in $v_3$ and $v_4$. Our proposed agreement relation $\sim_r$ addresses this loss of information by preserving the identity of the disagreeing agent. Specifically, $v_1$ and $v_6$ are grouped into block $B_1$, while $v_3$ and $v_4$ are grouped into $B_3$ by $\zeta(v)$ in Equation (8). Although both blocks correspond to the same 1-against-7 consensus level, $B_1$ indicates disagreement by Agent 2, whereas $B_3$ indicates disagreement by Agent 3.

**Algorithm 1** **C**onsistency-**D**riven **N**euro-**S**ymbolic **E**nsemble (CDNSE). Defaults used throughout the experiment: epochs $P = 10$, prior strength $\lambda = 2$, walk probability $p = 0.05$ (weighted) and 0.3 (warm), warm fraction $\rho = 0.05$, blocks probed per pass $B_M = 32$. WLS is the additive clause-weighting local search of Thornton et al. (2004); WALKSAT/SKC is adopted from (Selman et al., 1994).

**Require:** CNF $\mathcal{F}$; ensemble matrix $\mathbf{X} \in \{0,1\}^{M\times n}$; budget $B$
$\mathbf{w}, \kappa_w, c \leftarrow$ Level A; $\{B_1, \ldots, B_K\} \leftarrow$ Heider blocks ▷ $O(Nn)$, run once (§3.2)
$\mathbf{x} \leftarrow c$; **if** $\min_i E_i < E(c)$ **then** $\mathbf{x} \leftarrow \mathbf{x}_{i^\star}$, $i^\star = \arg\min_i E_i$ ▷ consensus seed
$\pi_v \leftarrow \lambda \cdot \mathrm{clip}\big(2\kappa_w(v) - 1,\ 0,\ 1\big)$; $\mathbf{cw} \leftarrow \mathbf{1}_m$; $u \leftarrow 0$
$\mathbf{x} \leftarrow$ WALKSAT/SKC$(\mathcal{F}, \mathbf{x}, \rho B)$; $u \leftarrow u + \rho B$
**for** $e = 1$ **to** $P$ **do**
  $\mathbf{x} \leftarrow \mathrm{WLS}\big(\mathcal{F}, \mathbf{x}, \lfloor (B-u)/P \rfloor;\ \mathbf{cw},\ \pi\big)$; $u \leftarrow u + \lfloor (B-u)/P \rfloor$
  **if** $E(\mathbf{x}) = 0$ **then return** $\mathbf{x}$
  stall $\leftarrow$ stall $+ 1$ if no improvement, else 0
  **if** stall $\geq 2$ **then**
    **for** $k$ in a random sample of $\min(B_M, K)$ blocks **do**
      $u \leftarrow u + |B_k|$; **if** $u > B$ **then break** ▷ probe charged to the budget
      **if** $E(\sigma_{B_k}(\mathbf{x})) < E(\mathbf{x})$ **then** $\mathbf{x} \leftarrow \sigma_{B_k}(\mathbf{x})$ ▷ Block flip (§3.3)
    stall $\leftarrow 0$
**return** the lowest-energy assignment seen

### 3.3 SEARCH BY BLOCK FLIPPING

Our proposed method is a *collaborative* approach that exploits model-consensus information, specifically through the two-level agreement introduced in §3.2. Rather than randomly selecting a subset of unfixed $k$ variables from $n$ variables, of which there are $\sum_{k=0}^{n} \binom{n}{k} = 2^n$ possible subsets, we select the subset according to the consensus among the models. Our proposed method is described in Algorithm 1.

Our formulation of agreement-informed variable blocks in §3.2.3 is inspired by balance theory, a well-established concept in social psychology that describes the formation of consistent relational structures within a group. The concept is originally proposed by Heider (1946) and given its graph-theoretic form by Cartwright & Harary (1956). Accordingly, we refer to these agreement-informed subsets as ***Heider blocks***, defined as

**Definition 1** (Heider blocks and phase flips)**.** The equivalence relation $(\sim_v, \sim_r)$ on $\mathcal{V}$ forms Heider blocks $B_1, \ldots, B_K$. For a block $B$, the phase flip $\sigma_B(\mathbf{x}) = \mathbf{x} \oplus \mathbf{1}_B$; i.e., inverting every variable of $B$ all at once.

We derive the upper bound for the number of Heider blocks given $M$ agents voting on $n$ variables:

**Theorem 2.** *The number of Heider blocks satisfies* $K \leq \min(n, 2^{M-1})$.

The established bound by Theorem 2 makes block flipping tractable, because it is independent of $n$ once $M$ is fixed. For the example of Table 1, $(N, n) = (8, 14)$ gives $K \leq \min(14, 2^7) = 14$ (for the largest instance in our later experimental results, $(M, n) = (8, 10410)$ gives $K \leq 128$), and the value we actually observe is $K = 115$. Our proposed phase flipping in Definition 1 chooses among $K \leq \min(n, 2^{M-1})$, that is, among 115 moves rather than the massive search space of the random flipping.

## 4 RESULTS

### 4.1 EXPERIMENTAL DESIGNS

#### 4.1.1 DATASETS AND PROTOCOL

**Agents.** Every instance carries the recorded best assignment of each of the $M = 8$ neural solvers, including: NEUROSAT (Selsam et al., 2019); the three G4SATBench literal–clause-graph backbones GGNN-LCG, GCN-LCG and GIN-LCG (Li et al., 2024); QUERYSAT (Ozoliņš et al., 2022); PDP (Amizadeh et al., 2019a); and the edge- and node-splitting CVIG variants of GMS (Liu et al.,

2021), GMS-E and GMS-N. The pool is deliberately heterogeneous in graph encoding (including literal–clause graph (lcg), variable–clause graph (vcg), and factor graph), in propagation depth (10–32 message-passing rounds), and in capacity (0.23–0.65 M parameters), but otherwise interchangeable.

**Training corpus.** All agents were first trained on a common corpus of 44,100 satisfiable SATLIB[2] instances spanning 46 uniform-random and controlled-backbone 3-CNF generator families. The corpus comprises `uf20-91`, `uf50-218`, `uf100-430`, and `uf200-860`; the forty `CBS_k3_n100_m_b` families with $m \in \{403, 411, 418, 423, 429, 435, 441, 449\}$ and backbone size $b \in \{10, 30, 50, 70, 90\}$; and `BMS_k3_n100_m429` and `RTI_k3_n100_m429`. The instances contain between 20 and 200 variables and between 91 and 860 clauses. The training corpus was divided into training and validation sets using an 80%–20% split, respectively. The model achieving the best validation score was retained as the final model. Training was performed for a maximum of 60 epochs, with early stopping applied after 15 consecutive epochs without improvement in the validation score.

**Testing corpus.** The trained agents are freed and make inference on an out-of-distribution test corpus, which is drawn from 6 classical structured families curated from SATLIB, including AIM, JNH, INDUCTIVE-INFERENCE, QG, textsc Bejing, and SSA. Ground truth comes from a complete CDCL solver under a $10^6$-conflict budget, which labelled every instance in the corpus: 123 satisfiable and 71 unsatisfiable. Solve rates (SR$^{\uparrow}$) are reported on the 123 satisfiable instances; the 71 refuted ones carry no solve-rate signal and are used for the soundness check instead; i.e., a method that ever reports a solution on one of them has a defect that no aggregate score would reveal.

### 4.1.2 BASELINE METHODS

**Fusion-based baseline.** A fusion operator is a fixed, parameter-free map $F(\mathbf{x}_1, \ldots, \mathbf{x}_N) \to \{0, 1\}^n$ that reads the ensemble matrix and emits one assignment, which is then polished by the local-search engine under the same budget. Writing $\mathrm{sc}(v) = \sum_i w_i \, \mathbf{x}_i[v]$ for a weight vector $\mathbf{w}$, a weighted majority sets $\mathbf{x}[v] = 1$ iff $\mathrm{sc}(v) > \frac{1}{2} \sum_i w_i$, ties resolved to the lowest-energy agent's value. We evaluate 11 fusion operators: MAXVOTE (weighted majority with $w \equiv 1$), AVGVOTE (mean bit thresholded at $\frac{1}{2}$), ENERGYWVOTE (weights of Equation (2)), CSSWVOTE ($w_i = (1 - E_i/m)^8$, the clause-satisfaction score sharpened), RANKWVOTE ($w_i = M - \mathrm{rank}(E_i)$), BESTOFN ($\arg\min_i E_i$), MEDOID (minimiser of $\sum_j d_{\mathrm{H}}(\mathbf{x}_i, \mathbf{x}_j)$), CLAUSEWVOTE (per-variable weights $W^4_{i,v}$, where $W_{i,v}$ is the fraction of $v$'s clauses that agent $i$ satisfies), ENTROPYVOTE (majority where the per-variable binary entropy is $\leq 0.9$, best agent elsewhere), GREEDYCONSENSUS (MAXVOTE then strict-descent repair, capped at $5n$ steps), and PAIRWISEXOVER (majority-of-three over all pairs with the best agent, lowest-energy child retained).

**Learning-based baseline.** We compare our proposed method with 10 learning-based ensemble methods, which is grouped into two categories.

The first class of methods is *formula-supervised without labels*. The formula itself is a supervision signal available at test time: every candidate assignment is scored by its energy, and every clause is a training example that an agent either satisfies or does not. Seven methods use only this information, including: BAGGING draws bootstrap samples of the $M$ agents, majority-votes within each bag and keeps the lowest-energy bag (Breiman, 1996), RANDSUBSPACE partitions the variables at random and lets each block be owned by the agent that satisfies most clauses touching it, the random-subspace method transposed to variables (Ho, 1998), ADABOOST runs Freund & Schapire (1997) with clauses as examples and agents as weak learners: at each round the agent with the lowest weighted unsatisfied mass receives voting weight $\alpha = \frac{1}{2} \ln\big((1 - \epsilon)/\epsilon\big)$ and the clauses it fails are up-weighted, GRADBOOST is the stagewise additive analogue: a real-valued score per variable to which agents are added with shrinkage while the thresholded score keeps reducing energy, BAYESAVG is Bayesian model averaging with the temperature of Equation (2) fitted rather than fixed, NEGCORR weights agents by accuracy times distinctness, rewarding diversity explicitly (Liu & Yao, 1999), and CLAUSEECOC lets each clause vote on its own variables with strength inversely proportional to how many agents already satisfy it.

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

Table 2: Benchmarking result on the 123 satisfiable instances, $B = 10^4$ flips, 10 seeds; the best member of each family of method. $\Delta$ and $p$ are paired against CDNSE over instances (Wilcoxon signed-rank). Superscript $^{\uparrow}$ means higher-is-better, $^{\downarrow}$ means lower-is-better. $^{\dagger}$median over runs that solved. BASELINE is the best agent without local search. PER-MODEL is the best agent with post-hoc local search. FUSION-based and LEARNING-based methods are described in §4.1.2.

| | | SR$^{\uparrow}$ | | median$^{\dagger}$ | | $\Delta$ vs. CDNSE | |
|---|---|---|---|---|---|---|---|
| **Method Class** | **Best Method** | % | 95% CI | **flips**$^{\downarrow}$ | **PAR-10**$^{\downarrow}$ | **(pp**$^{\uparrow}$**)** | $p$ |
| BASELINE | PDP | 11.4 | [6.5, 17.1] | – | 89.8k | +58.13 | $2.5\times10^{-16}$ |
| PER-MODEL | GIN-LCG | 60.9 | [53.3, 68.8] | 650 | 40.0k | +8.62 | 0.008 |
| FUSION | ClauseWVote | 60.2 | [52.3, 68.0] | 452 | 40.6k | +9.35 | 0.001 |
| LEARNING | StackGBM | 60.0 | [52.0, 67.8] | **358** | 40.7k | +9.51 | 0.002 |
| **CDNSE (Ours)** | | **69.5** | [61.8, 76.8] | 474 | **31.5k** | (ref.) | (ref.) |

The second class of methods is *cross-instance-supervised with labels from other instances*. Three ensemble methods are meta-learners over per-variable features; i.e., the $M$ agent bits plus occurrence count, polarity balance, mean bit and agreement mass, trained on the reference solutions of other instances and applied to the held-out set. These compared methods are STACKLOGREG, STACKFOREST and STACKGBM (Wolpert, 1992). Here, instances are partitioned into eight groups and the model applied to an instance is fitted only on the groups that exclude it, so no instance appears in its own training set.

### 4.2 EFFECTIVENESS ANALYSIS

**CDNSE outperforms baseline methods.** The recorded neural assignments are almost never solutions: the best agent solves 11.4% of the satisfiable instances outright. Combining all $M$ of them without any search barely moves this; i.e., the best of the 11 fusion operators reaches 12.2%, and the best of the 10 learned ensemble methods also reaches 12.2%, a gain of under one instance in both cases. Attaching a $10^4$-flip local search to the same assignments lifts them to 60.0 – 60.9%. The search contributes roughly 50 percentage points; the choice of what to combine contributes under two. CDNSE solves 69.5% of the satisfiable instances, against 60.2% for the best fusion operator (ClauseWVote) and 60.0% for the best learned combiner (StackGBM): +9.3 pp ($p = 0.001$) and +9.5 pp ($p = 0.002$), paired over instances. Against the strongest per-model polish (GIN-LCG) the gap is +8.6 pp ($p = 0.008$). Expected cost falls from 40.0k to 31.5k PAR-10 flips, a 21% reduction (Table 2). Over all 37 methods a Friedman test rejects equality ($p = 7.4\times10^{-27}$) and CDNSE and its ablations occupy the top of the mean-rank ordering.

**The advantage emerges at $10^4$ flips and persists through $10^8$ flips.** We evaluate method's performance across a five-order-of-magnitude range of flip budgets. At $10^3$ flips, the methods are not meaningfully separated: CDNSE achieves 40.9% compared with 40.7% for per-model polish, indicating that the consensus initialization and warm phase have not yet had sufficient budget to yield a measurable benefit. The gap emerges at $10^4$ flips (+8.9pp) and remains consistently positive thereafter: +4.5pp at $10^5$, +8.5pp at $10^6$, +7.3pp at $10^7$, and +7.3pp at $10^8$. At the largest budget, CDNSE solves 91.1% of instances, compared with 82.9% for per-model polish and 83.7% for the best fusion operator.

## 5 DISCUSSION

**The advantage does not arise from ensemble size.** A natural interpretation of the 9pp improvement over fusion is that CDNSE benefits simply from having access to a larger ensemble, with the advantage therefore expected to increase with $M$. Our results do not support this interpretation. For each $M' = 2, \ldots, 8$, we sample random sub-ensembles of size $M'$, reconstruct the consistency structure, and evaluate CDNSE against the best fusion operator. The performance gap ranges from 8.5 pp with two models to 10.4,pp with 8 models, remaining within 8.5–11.3 pp across all ensemble sizes and statistically significant throughout ($p \leq 0.002$). Thus, two agents already provide sufficient

disagreement to yield a measurable benefit, consistent with the effective ensemble size 2.1: additional independent models contribute limited information.

We vary the prior strength $\lambda$ and the number of consistency-annealed epochs $P$ independently around their default values. Across $\lambda \in [0, 8]$, the solve rate varies by at most 2.9 pp, while for $P \leq 20$ it remains within 66.5–69.6%, deteriorating only at $P = 40$ (62.8%). Every configuration, including $\lambda = 0$, outperforms the best fusion and learning baselines. The gain therefore appears robust to hyperparameter choice and is primarily attributable to the overall design of CDNSE. Notably, the comparable performance at $\lambda = 0$ indicates that the consensus prior itself is not responsible for the observed improvement under the present budget. The CDNSE initialization also reduces the starting energy by a factor of 27, with 0.62% versus 16.6% unsatisfied clauses for random initialization, while achieving statistically indistinguishable solve rates (68.6% versus 69.4%). Its advantage instead lies in search efficiency: the median number of flips is 482 versus 788 for random initialization, corresponding to a 1.64× reduction.

**Limitations.** We identify two limitations of the proposed method in its current form. First, the ensemble is not explicitly constructed to maximize model diversity. A deliberately diversified ensemble could increase $N_{\text{eff}}$ and potentially alter the relative benefits of combining and structuring. Second, the upper bound $2^{M-1}$ on the number of blocks is loose. Although this bound is useful because it is independent of the instance size $n$, at $M = 8$ it permits up to 128 blocks, whereas we observe at most 115 in our experiment. For substantially larger ensembles, the bound would eventually cease to provide a meaningful constraint, requiring the block structure to be coarsened explicitly, for example by imposing a minimum block size or by defining $\sim_r$ using a tolerance rather than exact equivalence.

**Future work.** Two directions follow naturally from the proposed framework. First, the block decomposition is currently computed once from the initial ensemble. Recomputing it from the incumbent assignments would instead yield a dynamic consistency structure that evolves with the search trajectory, potentially allowing the method to adapt its variable blocks as the ensemble's predictions change. Second, the same algebra extends beyond SAT to other constraint problems, including MaxSAT, model counting, and scheduling problems, provided that multiple trained discrete predictors produce competing complete assignments over a shared variable set. In this broader setting, agreement among predictors can likewise be converted into a structured decomposition of the underlying decision variables.

## 6 CONCLUSION

The key signal in an ensemble is not how much its members disagree, but how their disagreements are organised. Our proposed consistency structure captures these relationships and partitions variables into coherent blocks that directly define the moves of local search. This transforms an otherwise combinatorial search over individual variables into a substantially smaller block-level search. Empirically, the structure primarily improves search efficiency rather than initial solution quality: our seeding strategy substantially reduces the initial energy, while the main benefit is fewer flips to solution.

The framework is not specific to SAT solver and can, in principle, apply whenever multiple predictors produce discrete assignments over shared variables. This includes binary or quantised neural representations and potentially intermediate commitments in large language models, where consistency could identify shared decisions and groups of decisions that tend to change together. However, extensions beyond SAT require an appropriate verification objective; the reliability of the resulting search ultimately depends on the quality of that objective.

### AI USE STATEMENT

In this work, we used generative AI tools for grammar and language editing, code and theorem auditing, and simulated peer review of the original draft. We did not use generative AI tools for method formulation, implementation planning, or the formulation or proof of theorems. We have carefully reviewed all AI-assisted work and take full responsibility for the final content of this work, including all text, claims, analyses, and other artifacts produced with the assistance of generative AI.

### Ethics statement

We declare that there are no conflicts of interest.

### Reproducibility Statement

We provide detailed descriptions of the proposed method and its implementation in the main text and appendices. The source code implementing our method is also submitted with the manuscript to facilitate reproducibility.

## References


Saeed Amizadeh, Sergiy Matusevych, and Markus Weimer. Learning to solve circuit-SAT: an unsupervised differentiable approach. In *International Conference on Learning Representations (ICLR)*, 2019a.

Saeed Amizadeh, Sergiy Matusevych, and Markus Weimer. PDP: a general neural framework for learning constraint satisfaction solvers. *arXiv preprint arXiv:1903.01969*, 2019b.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1007/BF00058655.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Shaowei Cai and Kaile Su. Local search for Boolean satisfiability with configuration checking and subscore. *Artificial Intelligence*, 204:75–98, 2013. doi: 10.1016/j.artint.2013.09.001.

Chris Cameron, Rex Chen, Jason Hartford, and Kevin Leyton-Brown. Predicting propositional satisfiability via end-to-end learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3324–3331, 2020. doi: 10.1609/aaai.v34i04.5733.

Dorwin Cartwright and Frank Harary. Structural balance: a generalization of Heider's theory. *Psychological Review*, 63(5):277–293, 1956. doi: 10.1037/h0046049.

Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC)*, pp. 151–158, 1971. doi: 10.1145/800157.805047.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, Lecture Notes in Computer Science, pp. 1–15. Springer, 2000. doi: 10.1007/3-540-45014-9_1.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995. doi: 10.1613/jair.105.

Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing (SAT)*, Lecture Notes in Computer Science, pp. 502–518. Springer, 2004. doi: 10.1007/978-3-540-24605-3_37.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Chengrui Gao, Haopu Shang, Ke Xue, and Chao Qian. Neural solver selection for combinatorial optimization. *arXiv preprint arXiv:2410.09693*, 2024.

Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946. doi: 10.1080/00223980.1946.9917275.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. doi: 10.1109/34.709601.

Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–401, 1999. doi: 10.1214/ss/1009212519.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 231–238, 1994.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003. doi: 10.1023/A:1022859003006.

Zhaoyu Li, Jinpei Guo, and Xujie Si. G4SATBench: benchmarking and advancing SAT solving with graph neural networks. *Transactions on Machine Learning Research*, 2024.

Minghao Liu, Fuqi Jia, Pei Huang, Fan Zhang, Yuchen Sun, Shaowei Cai, Feifei Ma, and Jian Zhang. Can graph neural networks learn to solve maxsat problem? *arXiv preprint arXiv:2111.07568*, 2021.

Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10): 1399–1404, 1999. doi: 10.1016/S0893-6080(99)00073-8.

Yuri Malitsky, Ashish Sabharwal, Horst Samulowitz, and Meinolf Sellmann. Parallel sat solver selection and scheduling. In *International Conference on Principles and Practice of Constraint Programming*, pp. 512–526. Springer, 2012.

João Marques-Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2nd edition, 2021. doi: 10.3233/FAIA200987.

João P. Marques-Silva and Karem A. Sakallah. GRASP: a search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, 1999. doi: 10.1109/12.769433.

David Mojžíšek, Jan Hla, Ziwei Li, Ziyu Zhou, and Mikoláš Janota. Neural approaches to sat solving: Design choices and interpretability. *International Journal of Approximate Reasoning*, pp. 109609, 2025.

Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC)*, pp. 530–535, 2001. doi: 10.1145/378239.379017.

Emils Ozoliņš, Karlis Freivalds, Andis Draguns, Eliza Gaile, Ronalds Zakovskis, and Sergejs Kozlovics. Goal-aware neural SAT solver. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9892733.

Marcelo Prates, Pedro H. C. Avelar, Henrique Lemos, Luis C. Lamb, and Moshe Y. Vardi. Learning to solve NP-complete problems: a graph neural network for decision TSP. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4731–4738, 2019. doi: 10.1609/aaai.v33i01.33014731.

Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI)*, pp. 337–343, 1994.

Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. Learning a SAT solver from single-bit supervision. In *International Conference on Learning Representations (ICLR)*, 2019.

John Thornton, Duc Nghia Pham, Stuart Bain, and Valnir Ferreira Jr. Additive versus multiplicative clause weighting for SAT. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, pp. 191–196, 2004.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Zhanguang Zhang, Didier Chételat, Joseph Cotnareanu, Amur Ghose, Wenyi Xiao, Hui-Ling Zhen, Yingxue Zhang, Jianye Hao, Mark Coates, and Mingxuan Yuan. Grass: Combining graph neural networks with expert knowledge for sat solver selection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6301–6311, 2024.

# A SUPPLEMENTAL FIGURES

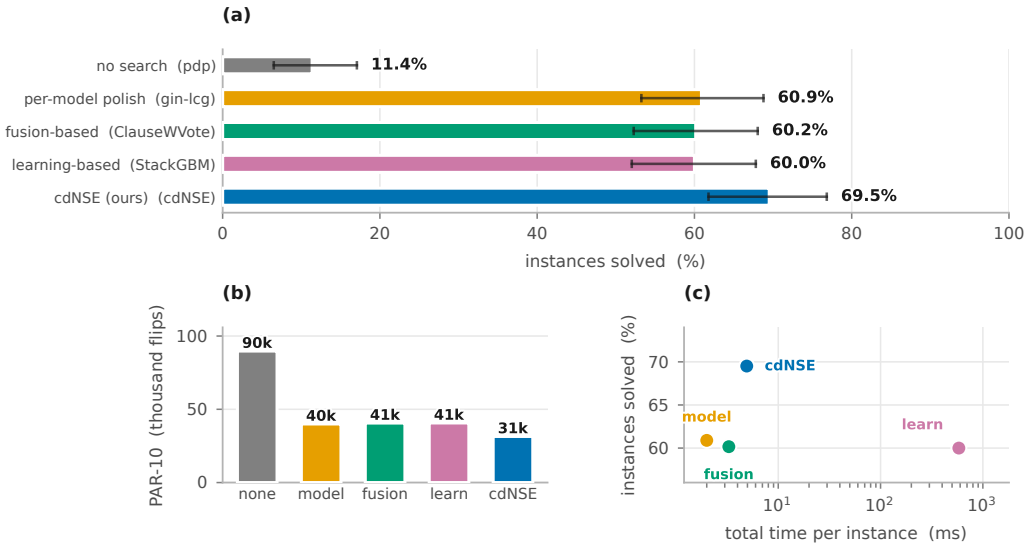



Figure A.1: Summary of experimental results in §4. **(a)** SR$^{\uparrow}$ with bootstrap intervals over instances. **(b)** PAR-10$^{\downarrow}$, the expected flips per instance when unsolved runs are charged $10B$. **(c)** The cost–benefit trade-off across method families: learned ensemble methods incur the highest computational cost while failing to deliver a commensurate improvement in performance.

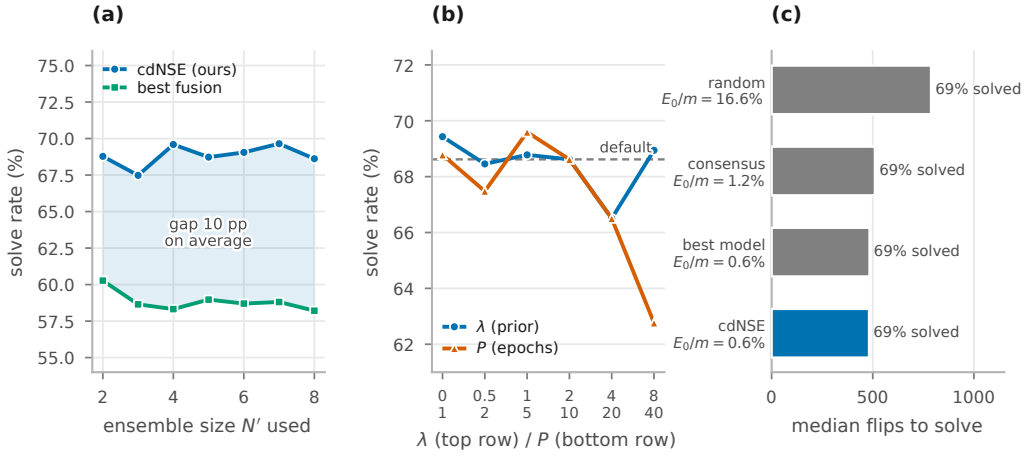



Figure A.2: **(a)** The advantage over the best fusion operator does not need a large ensemble: it is already 8.5 pp at $M' = 2$ and does not grow with $M'$. **(b)** Sweeping the consensus-prior strength $\lambda$ and the epoch count $P$ around their defaults. **(c)** Seeding rules differ by $27\times$ in the energy they start from and not at all in solve rate.

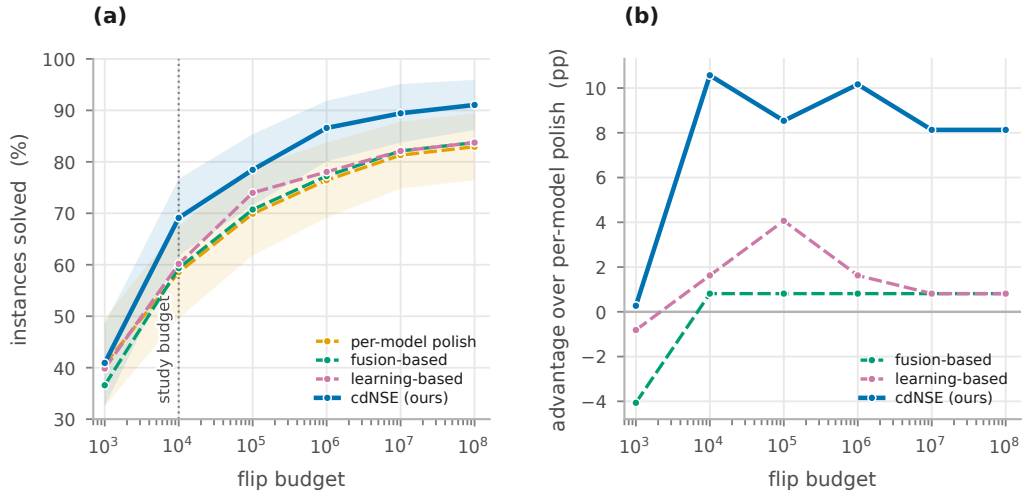



Figure A.3: **Anytime behaviour from $10^3$ to $10^8$ flips. (a)** Solve rate against budget. **(b)** Advantage over per-model polish.

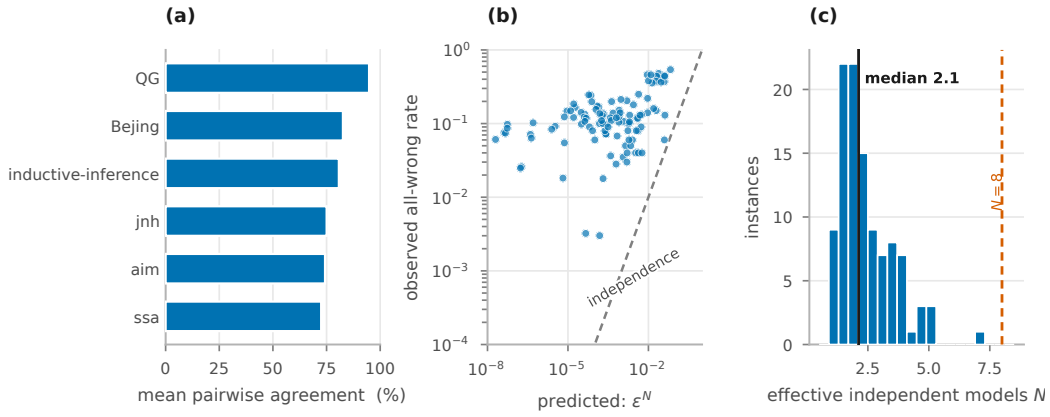



Figure A.4: **Why combination has so little to offer. (a)** The agents agree with one another on 77.7% of variables on average. **(b)** The rate at which all agents are simultaneously wrong, against the rate that would follow if their errors were independent; every instance lies above the diagonal. **(c)** The implied number of effective independent agents.

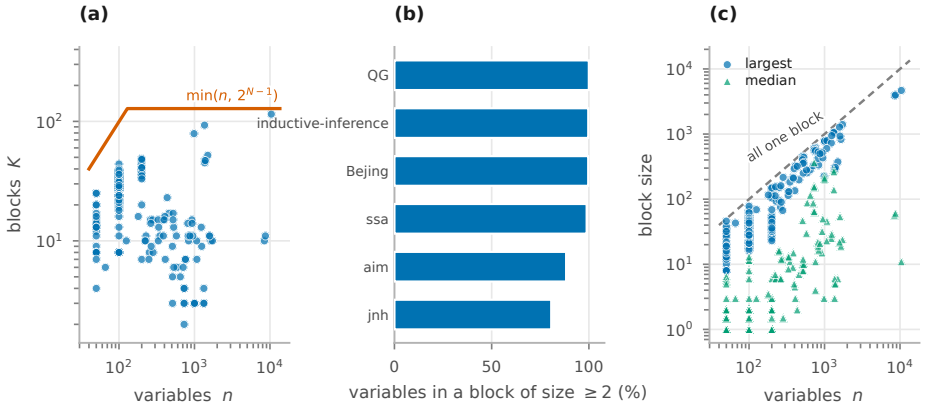



Figure A.5: Summary of Heider block size computed from the testing corpus Table B.1. This confirms Theorem 2.

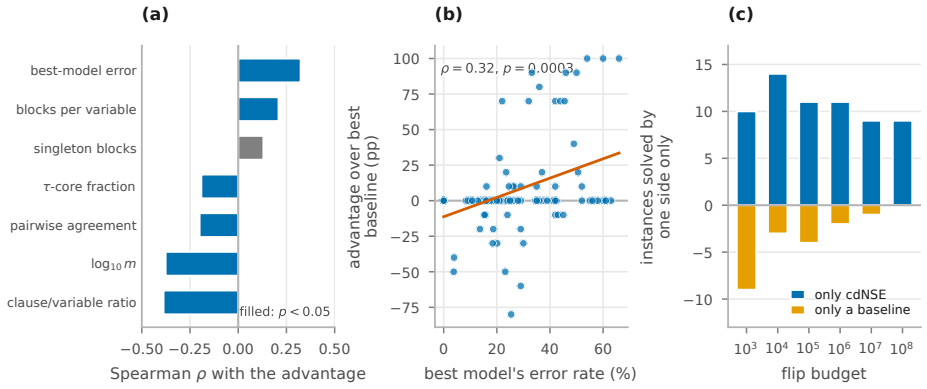



Figure A.6: **Where the advantage lives.** **(a)** Spearman correlation of CDNSE's per-instance advantage over the strongest baseline against quantities measurable *before* any search is run. **(b)** The advantage grows with the error rate of the best individual model. **(c)** Instances one side solves and the other does not, at each budget: the count of instances only a baseline reaches falls to zero, while the count only CDNSE reaches does not.

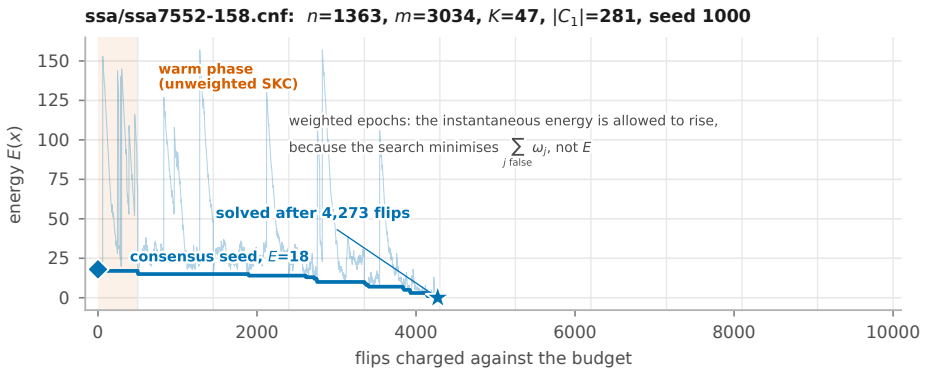



Figure A.7: Local search trajectory of an instance.

## B SUPPLEMENTAL TABLES

Table B.1: Summary of the testing corpus in §4.

| Family | inst. | SAT | UNSAT | $n$ range | $m$ range |
|---|---|---|---|---|---|
| BEJING | 10 | 9 | 1 | 125–8704 | 310–47820 |
| QG | 13 | 5 | 8 | 512–1331 | 9685–64054 |
| AIM | 72 | 48 | 24 | 50–200 | 80–1200 |
| INDUCTIVE-INFERENCE | 41 | 41 | 0 | 66–1728 | 186–24792 |
| JNH | 50 | 16 | 34 | 100–100 | 800–900 |
| SSA | 8 | 4 | 4 | 435–10410 | 1027–34238 |
| **total** | **194** | **123** | **71** | 50–10410 | 80–64054 |

Table B.2: Results of 11 **fusion-based** baselines in Section 4.1.2, each polished under $B = 10^4$ flips and 10 seeds, sorted by SR. †median over runs that solved.

| Method | solve % | 95% CI | median flips† | PAR-10 | Δ vs. CDNSE (pp) | $p$ |
|---|---|---|---|---|---|---|
| ClauseWVote | 60.2 | [52.3, 68.0] | 452 | 40.6k | +9.35 | 0.001 |
| RankWVote | 59.9 | [51.9, 67.7] | 385 | 40.8k | +9.59 | 0.002 |
| MaxVote | 59.2 | [51.1, 67.4] | 368 | 41.5k | +10.33 | $9.1\times10^{-4}$ |
| AvgVote | 59.1 | [50.9, 67.1] | 360 | 41.6k | +10.41 | $5.2\times10^{-4}$ |
| EntropyVote | 59.0 | [51.0, 67.1] | 342 | 41.6k | +10.49 | $2.8\times10^{-4}$ |
| GreedyConsensus | 59.0 | [51.1, 66.8] | 346 | 41.7k | +10.49 | $3.3\times10^{-4}$ |
| Medoid | 58.6 | [50.7, 66.6] | 475 | 42.1k | +10.89 | $2.2\times10^{-4}$ |
| EnergyWVote | 58.5 | [50.5, 66.6] | 338 | 42.1k | +10.98 | $9.2\times10^{-5}$ |
| CSSWVote | 58.5 | [50.4, 66.4] | 409 | 42.2k | +11.06 | $2.3\times10^{-4}$ |
| BestOfN | 58.4 | [50.2, 66.3] | 288 | 42.2k | +11.14 | $8.6\times10^{-5}$ |
| PairwiseXover | 58.3 | [50.2, 66.2] | 321 | 42.3k | +11.22 | $3.5\times10^{-5}$ |

Table B.3: Results of 10 **learning-based** baselines of Section 4.1.2. The three STACK ensemble methods are cross-fitted meta-learners that see reference solutions of other instances; the rest are supervised by the formula alone.

| Method | solve % | 95% CI | median flips† | PAR-10 | Δ vs. CDNSE (pp) | $p$ |
|---|---|---|---|---|---|---|
| StackGBM | 60.0 | [52.0, 67.8] | 358 | 40.7k | +9.51 | 0.002 |
| StackForest | 59.5 | [51.5, 67.3] | 320 | 41.1k | +10.00 | $6.2\times10^{-4}$ |
| BayesAvg | 59.5 | [51.1, 67.6] | 306 | 41.2k | +10.00 | $3.7\times10^{-4}$ |
| ClauseECOC | 59.3 | [51.4, 67.1] | 404 | 41.4k | +10.24 | $8.0\times10^{-4}$ |
| NegCorr | 58.8 | [50.6, 66.7] | 282 | 41.8k | +10.73 | $1.0\times10^{-4}$ |
| Bagging | 58.3 | [50.2, 66.3] | 314 | 42.3k | +11.22 | $2.2\times10^{-5}$ |
| GradBoost | 58.3 | [50.2, 66.3] | 303 | 42.4k | +11.22 | $2.4\times10^{-5}$ |
| RandSubspace | 58.3 | [50.3, 66.3] | 304 | 42.4k | +11.22 | $2.4\times10^{-5}$ |
| StackLogReg | 58.0 | [50.2, 65.9] | 452 | 42.7k | +11.46 | $3.5\times10^{-4}$ |
| AdaBoost | 57.7 | [49.7, 65.7] | 316 | 42.9k | +11.79 | $1.1\times10^{-4}$ |

Table B.4: Per-model local search: each of the $M$ neural solvers' own assignment used directly as the seed of WALKSAT/SKC.

| Method | solve % | 95% CI | median flips† | PAR-10 | Δ vs. CDNSE (pp) | $p$ |
|---|---|---|---|---|---|---|
| GIN-LCG | 60.9 | [53.3, 68.8] | 650 | 40.0k | +8.62 | 0.008 |
| GGNN-LCG | 60.6 | [52.5, 68.5] | 509 | 40.2k | +8.94 | 0.005 |
| GMS-E | 60.4 | [52.7, 67.9] | 459 | 40.3k | +9.11 | 0.005 |
| PDP | 60.3 | [52.4, 68.3] | 364 | 40.4k | +9.19 | 0.001 |
| NEUROSAT | 58.8 | [50.7, 66.9] | 489 | 41.9k | +10.73 | $3.0\times10^{-4}$ |
| GCN-LCG | 58.8 | [51.1, 66.7] | 379 | 42.0k | +10.73 | $1.6\times10^{-4}$ |
| QUERYSAT | 58.5 | [50.3, 66.6] | 450 | 42.3k | +11.06 | $1.1\times10^{-4}$ |
| GMS-N | 57.6 | [49.7, 65.4] | 453 | 43.1k | +11.87 | $8.5\times10^{-5}$ |

Table B.5: Ensemble diversity and error correlation, family by family. $\bar{\varepsilon}$ is the mean per-agent error rate against the CDCL reference, $\varepsilon_{\min}$ the best agent's and $\varepsilon_{\text{cons}}$ the consensus's; "all-wrong" is the fraction of variables on which every agent errs. All rates are measured against the CDCL reference solution of each satisfiable instance.

| Family | pairwise agree. | $\lvert C_1\rvert/n$ | $\bar{\varepsilon}$ | $\varepsilon_{\min}$ | $\varepsilon_{\text{cons}}$ | all-wrong |
|---|---|---|---|---|---|---|
| BEJING | 0.826 | 0.483 | 0.263 | 0.139 | 0.266 | 0.079 |
| QG | 0.948 | 0.792 | 0.119 | 0.098 | 0.099 | 0.079 |
| AIM | 0.743 | 0.329 | 0.466 | 0.315 | 0.451 | 0.154 |
| INDUCTIVE-INFERENCE | 0.807 | 0.481 | 0.316 | 0.192 | 0.282 | 0.119 |
| JNH | 0.750 | 0.365 | 0.386 | 0.286 | 0.383 | 0.102 |
| SSA | 0.725 | 0.361 | 0.499 | 0.433 | 0.482 | 0.121 |

Table B.6: This table summarizes the computational cost incurred by each method family relative to the performance it delivers. The "setup" cost denotes the one-time, corpus-level expense incurred before deployment. Only the cross-instance meta-learning methods incur such a cost, as they require cross-fitting on reference solutions from other instances. The "ensemble method" cost denotes the per-instance expense of converting the ensemble outputs into a single initial assignment, while "total" denotes the overall per-instance cost of the deployed method, including both combination and subsequent search. CDNSE incurs no separate setup cost and constructs its structure internally during solving; consequently, the cost of structure construction is already included in its reported total.

| | | cost | | | benefit | |
|---|---|---|---|---|---|---|
| Family of method | Best member | setup (s) | combiner (ms) | total (ms) | solve % | PAR-10 |
| PER-MODEL | gin-lcg | – | 0.00 | 2.0 | 60.9 | 40.0k |
| FUSION | ClauseWVote | – | 1.12 | 3.3 | 60.2 | 40.6k |
| LEARNING | StackGBM | 291 | 580.58 | 582.6 | 60.0 | 40.7k |
| **CDNSE** | | – | 0.00 | 4.9 | 69.5 | 31.5k |

Table B.7: **Ensemble-size robustness.** For each $M'$, random sub-ensembles of that size are drawn, the consistency structure is rebuilt from the subset alone, and both methods are re-run on it. $p$ is a paired Wilcoxon signed-rank test over instances. Seeds 3000–3004, disjoint from the main evaluation.

| $N'$ | best fusion % | CDNSE % | Δ (pp) | $p$ |
|---|---|---|---|---|
| 2 | 60.3 | 68.8 | +8.5 | 0.001 |
| 3 | 58.6 | 67.5 | +8.8 | 0.002 |
| 4 | 58.3 | 69.6 | +11.3 | $2.2\times10^{-5}$ |
| 5 | 59.0 | 68.7 | +9.8 | $5.8\times10^{-4}$ |
| 6 | 58.7 | 69.1 | +10.4 | $5.8\times10^{-5}$ |
| 7 | 58.8 | 69.6 | +10.8 | $1.8\times10^{-5}$ |
| 8 | 58.2 | 68.6 | +10.4 | $6.6\times10^{-4}$ |

# C Proofs of Theorems

## C.1 Proof of Theorem 1

Write $S = \{\mathbf{0}_M,\ \mathbf{1}_M\} \subseteq \{0,1\}^M$, so that Equation (6) reads $u \sim_r v \iff (u,v) \in S$. Two observations suffice:

1. Componentwise, $\mathbf{x}_i[v] \oplus \mathbf{x}_i[v] = 0$ and $\oplus$ is associative and commutative, so for all $u, v, w \in \mathcal{V}$ and every $i \in \mathcal{A}$,
$$\delta_i(u,u) = 0, \quad \delta_i(u,v) = \delta_i(v,u), \quad \delta_i(u,v) \oplus \delta_i(v,w) = \mathbf{x}_i[u] \oplus \mathbf{x}_i[w] = \delta_i(u,w), \tag{10}$$
and therefore, stacking over $i$, $(u,u) = \mathbf{0}_M$, $(u,v) = (v,u)$ and $(u,w) = (u,v) \oplus (v,w)$.
2. $S$ is a subgroup of $\big(\{0,1\}^M, \oplus\big)$. It contains the identity $\mathbf{0}_M$ and is closed, since $\mathbf{0}_M \oplus \mathbf{0}_M = \mathbf{1}_M \oplus \mathbf{1}_M = \mathbf{0}_M$ and $\mathbf{0}_M \oplus \mathbf{1}_M = \mathbf{1}_M$; every element is its own inverse.

Reflexivity is then $(u,u) = \mathbf{0}_M \in S$; symmetry is $(u,v) = (v,u)$; and transitivity is $(u,w) = (u,v) \oplus (v,w) \in S \oplus S \subseteq S$.

## C.2 Proof of Theorem 2

$u \sim_r v$ iff $z(u) \in \{z(v), \bar{z}(v)\}$, so the blocks are exactly the fibres of the canonical code $\zeta = \min(z, \bar{z})$ of Equation (8) and $K = |\zeta(\mathcal{V})|$. The map $c \mapsto \{c, (2^M - 1) - c\}$ pairs the $2^M$ codes into $2^{M-1}$ unordered pairs, and no pair being a singleton because $c = (2^M - 1) - c$ has no solution in the integers ($2^M - 1$ is odd). The canonical code is constant on each pair and distinct across pairs, so it takes at most $2^{M-1}$ values. It is also a function on $\mathcal{V}$, hence takes at most $n$ values. Therefore $K \leq \min(n, 2^{M-1})$.