AGENT-GUIDED HYPOTHESIS-DRIVEN SYMBOLIC REGRESSION FOR CAR-FOLLOWING DYNAMICS DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Traffic congestion remains a persistent problem for modern cities, and the growing presence of autonomous vehicles has made understanding driving behaviors more essential than ever. Car-following models are fundamental to analyzing these dynamics. Their formulas have traditionally been derived by professionals guided by physical intuition and data fitting, yielding models that are both interpretable and transferable. Discovering such a model from trajectory data follows the classic scientific loop: generating a hypothesis, collecting data, analyzing it, validating the hypothesis, and repeating. The steps from analysis, to validation, to regeneration are professionally driven but labor-intensive, often demanding many rounds of validating and revising the hypothesis. This paper asks whether these steps can be automated. We propose agent-guided hypothesis-driven symbolic regression (AHSR), a framework that couples a large language model (LLM) with a deep reinforcement-learning symbolic regression (DRL-SR) engine to discover car-following dynamics directly from vehicle trajectories. Informed by car-following literature and an initial skim of the trajectory data, the LLM proposes a hypothesis template that specifies the hypothesis structure, the candidate variables with their bounds, and the physical units. DRL-SR then explores expression candidates within this template, and the LLM revises the hypothesis in light of DRL-SR's findings and returns an updated template. This iteration repeats until a concise and explainable expression is found, ensuring that every explored expression is physically feasible, explainable, and accurate. Experiments show that AHSR is capable to recover classical and most widely-used car-following models on synthetic data with noise. These results suggest that AHSR can automate the discovery of explainable car-following models, reducing the manual effort of the traditional scientific loop while preserving physical interpretability.

1 INTRODUCTION

Car-following models describe how a driver adjusts longitudinal acceleration in response to the vehicle ahead. Before or around 2000, researchers derived car-following model benchmarks such as General Motors model (Chandler et al., 1958), the GHR model (Gazis et al., 1959; 1961), and the Intelligent Driver Model (IDM) (Treiber et al., 2000) through a scientific process paradigm that develops model structure terms by experts with a reasonable hypothesis and then validates the hypothesis afterwards. This paradigm usually includes observing driving behavior, reasoning from physical principles, proposing hypothesis, describing hypothesis with mathematical expressions, fitting coefficients with data, revising hypothesis or structures until a compact, interpretable model emerged. Once a hypothesis and the mathematical form are ready, calibrating its coefficients to a given dataset is comparatively routine. The lasting difficulty, then, is hypothesis and mathematical form derivation, in other words, is arriving at a functional structure that is compact, physically meaningful, and accurate enough to the observed dynamics. This burden has only grown whenever the prevailing dynamics shift under driver heterogeneity or the emerging mixed traffic of automated and human-driven vehicles. The existing forms may no longer suffice, and a new derivation may be required rather than a mere recalibration. What is ultimately at stake is therefore a better form, not merely a better fit. This makes model building a discovery problem, where an automated frame-

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.

work that retrace that same process of reasoning from established theory to a candidate model and calibrating it with data may be the solution. On the other hand, in the scientific model deriving process, the steps of building to revising the hypothesis carry most of the intellectual weight. Bridging these steps is a long sequence of designs, checking, and corrections carried out manually, which is exactly this expert-intensive bridge that an automated discovery framework seeks to shoulder. The classic car-following models were developed in an era dominated by human-driven vehicles (HDVs), but the arrival of advanced driver-assistance systems (ADAS), autonomous vehicles (AVs), and connected automated vehicles (CAVs) is altering how vehicles respond to one another, such as CAV-to-CAV as well as CAV-to-HDV, that the classic models were not designed to capture (Zhong et al., 2020; Xiao et al., 2018). Meanwhile the supply of trajectory data has exploded, streaming in from onboard sensing and from roadside infrastructure such as loop detectors and intersection cameras (Gloudemans et al., 2023), and thus processing datasets of this magnitude manually has become impractical. This is why an automated pipeline deriving models that stay compact, physically legible, and transferable across settings has become attractive. As previous works (Tenachi et al., 2023; Angah et al., 2024) mentioned similarly, this research argues that a model that deserves confidence should balance among four following fitting objectives. **Parsimony** favors the compact expression that still accounts for the data, resisting the temptation to pile on terms or tokens. **Interpretability** demands that each symbol and operation carry legible physical meanings. This is an especially valuable trait when deriving unknown dynamics. **Generalizability** asks that a form fitted in one setting continue to hold when it meets vehicles, locations, and conditions it was never calibrated. **Accuracy**, finally, gauges how tightly the model's outputs track what was actually observed, typically through measures such as mean squared error. Because these criteria can trade off against one another, a discovery method has to consider all four together. Symbolic regression (SR) searches the open-ended space of mathematical expressions that explains the data. The difficulty is that the SR search space is enormous. The number of syntactically valid expressions grows exponentially with expression length (Mundhenk et al., 2021), and empirical benchmarks (La Cava et al., 2021) show that the accuracy of state-of-the-art SR methods degrades sharply as the target expression becomes longer and more nested. The IDM is a stark example. The structure of IDM comprises more than twenty tokens combining ratios, powers, and a square root over three input variables, placing it beyond the reach of the basic SR that attempts to recover the entire equation correctly. Genetic programming (GP) evolves a population of expression trees through mutation and crossover (Koza, 1994; Schmidt & Lipson, 2009), and remains competitive in modern tools such as PySR (Cranmer et al., 2020). Deep-learning approaches frame SR as a sequential decision making task. Petersen et al. (2019) train an autoregressive policy to emit prefix-notation token sequences under a risk-seeking policy gradient, and Mundhenk et al. (2021) combine this neural network explorer with a GP inner loop. Transformer-based methods instead pre-train on massive synthetic equation corpora to map data directly to expressions (Biggio et al., 2021). A particularly relevant advance is PhySO (Tenachi et al., 2023), which enforces physical unit analysis as a hard constraint. These methods have steadily pushed the frontier of recoverable expression complexity, yet all of them share a common ceiling that recovering a complex model such as IDM remains challenging. Increasing compute or population size yields diminishing returns because the fundamental obstacle is the size of the space being searched. This motivates a change of strategy, from searching harder to searching smarter. A human scientist confronted with a complex structure does not enumerate expressions blindly, instead they reason from established theory, decompose the problem into physically meaningful parts, and solve each part in turn. Two recent developments make it possible to mechanize this reasoning. First, large language models (LLMs) have absorbed a vast body of scientific literature and can generate, critique, and revise structured hypotheses. They have already been used to seed genetic populations (Liu et al., 2024), select among candidate expressions (Shojaee et al., 2024), and drive verifiable discovery loops such as FunSearch (Romera-Paredes et al., 2024). Second, retrieval-augmented generation (RAG) (Lewis et al., 2020) lets an LLM query an external document collection for providing verifiable facts. Benefits of RAG are also demonstrated in scientific settings from materials prediction to theorem proving (Yang et al., 2023). We propose the Agent-Guided Hypothesis-Driven Symbolic Regression (AHSR) framework, an automated framework that discovers car-following dynamics directly from trajectory data by mimicking the scientific process of human researchers. AHSR couples a large language model (LLM) acting as a *hypothesis architect*, learning past car-following theory, with a DRL-SR method acting as the *explorer*. The LLM proposes a physically meaningful decomposition of the target function into a few short portions, an assembly template that recombines them, and a physical unit consistency argument, while the DRL-SR engine searches each portion. Then the LLM refines the hypothesis based on

the exploration, closing the loop with a hypothesize, test, refine cycle. The framework is designed to jointly target parsimony, interpretability, accuracy, and generalizability. Each portion is a compact sub-problem solved independently by a physical unit aware DRL-SR search. Because search cost grows exponentially with expression length, replacing one long search with several short ones shrinks the problem dramatically, bringing each sub-task within the reach of contemporary DRL-SR while keeping the resulting expressions concise and physically feasible. We validate AHSR on synthetic traffic trajectory datasets generated from GM and IDM models.

2 Literature Review

Early stimulus-response Car-following formulations can be traced to the GM (Chandler et al., 1958) model and the GHR model (Gazis et al., 1961). These models provided a physically tractable and concise mechanism of traffic dynamics with empirical data. IDM (Treiber et al., 2000) describes three basic driving situations with one continuous equation: accelerating from rest, following a preceding vehicle, and approaching a slow vehicle, where every term carries a clear physical meaning, and thus a single continuous equation to yield collision-free behavior across a wider range of traffic regimes. SR searches the space of mathematical expressions to find a formula that best describes observed data. Previous approaches based on genetic programming (GP) (Koza, 1994; Schmidt & Lipson, 2009) established SR as a tool for discovering mathematical equation. The strength of GP is its flexibility. It does not require a fixed functional structure and can, in principle, assemble arbitrary expression trees. . This same freedom, however, makes token sampling inefficient, and its performance becomes unstable when the target expression is long or the data are noisy. The emerging transformer-based and DRL-SR methods substantially improved search efficiency of expression recovery. Petersen et al. (2019) cast SR as a sequential decision process in which an autoregressive policy samples prefix-notation token sequences and is trained by a risk-seeking policy gradient. Mundhenk et al. (2021) extended this approach by using the learned policy to seed a genetic-programming population, combining neural guidance with evolutionary local search. Biggio et al. (2021) pre-trained a transformer on a large dataset of synthetic equations to map numeric data directly to expressions. PhySO (Tenachi et al., 2023) incorporated physical unit analysis as a hard constraint, biasing the search toward physically plausible expressions and improving sample efficiency in multi-variable physics problems. Compared with pure GP, the risk seeking policy search is more efficiently and, in PhySO's case, deploying dimensional constraints prunes implausible candidates (Tenachi et al., 2023). Their efficiency nonetheless faces a limit when ground-truth expression is complicated because the search space grows quickly with expression length (Mundhenk et al., 2021) and recovery accuracy degrades as ground-truth complexity increases (La Cava et al., 2021), especially expressions with the length and nested structure of a full car-following model, such as IDM, remain difficult to recover directly. LLMs have demonstrated the ability to reason input texts and generate responses. Romera-Paredes et al. (2024) demonstrated this with FunSearch, in which an LLM iteratively proposes candidate computational language programs that an automated evaluator scores and feeds back to seed the next round. Bran et al. (2024) coupled GPT-4 with a chemistry robotic platform to plan synthetic routes and physically synthesize target molecules, while Boiko et al. (2023) built a GPT-4-driven agent that automatically designs experiments, writes Python scripts to operate robots, and iteratively optimizes reaction conditions from the results it measures. Cox et al. (2024) used LLMs to propose mathematical equations from noisy data to downstream SR verification. Within SR specifically, LLM-augmented approaches have been used to seed genetic populations to select among candidate expressions (Liu et al., 2024; Shojaee et al., 2024), and to generate prior that is physical unit consistent (Tenachi et al., 2023). The advantage of these methods is that they inject prior scientific knowledge encoded in the LLM into the search, improving both exploration efficiency and the physical plausibility of candidates. RAG is demonstrated to be able to supply verifiable facts. It has been applied for proving mathematical theorems (Yang et al., 2023). Lewis et al. (2020) pairs a language model with a searchable document library, letting the model look up relevant material and answer based on the library. SR-LLM (Guo et al., 2025) introduced an incremental SR loop with an LLM that retrieves prior expressions and prior exploration knowledge to propose expression refinement. These approaches confirm that conditioning symbolic search on retrieved knowledge improves both exploration efficiency and physical plausibility. However, they retrieve knowledge to guide token-level search rather than to propose an explicit, human-interpretable hypothesis of the potential governing equation. AHSR fills this gap by having an LLM architect retrieve published past theories from a RAG knowledge base

and propose an explicit, human-interpretable decomposition of the governing equation law. Operating at the level of structural hypotheses rather than individual tokens lets the retrieved knowledge shape the global form of the model instead of merely biasing the next token, thereby mimicking the way a human scientist reasons from established theory toward the discovery of new dynamics. Their limitation is that the LLM still operates at the level of individual tokens or candidate expressions. It biases which symbols appear, which candidates survive, or which symbols in the bucket for selecting, but it never commits to an explicit, human-interpretable hypothesis about the structural form of the potential governing equation. The challenge of discovering complex physical models from data is well-recognized in the SR literature. The size of the search space grows exponentially with expression length (Mundhenk et al., 2021), and empirical benchmarking shows that the accuracy of SR methods degrades sharply as the ground truth's complexity grows (La Cava et al., 2021). To mitigate this, AI Feynman (Udrescu et al., 2020a) recursively simplifies a hard regression problem by fitting a neural-network surrogate to the data to split the governing equation target into lower-dimensional sub-functions that are individually easier to recover. AI Feynman 2.0 (Udrescu et al., 2020b) extends this idea by detecting graph modularity and decomposing the expression into modular sub-graphs that are solved separately and recombined along a Pareto accuracy-complexity frontier. Champion et al. (2019) jointly train an autoencoder and a Sparse Identification of Nonlinear Dynamics (SINDy) model so that a parsimonious governing equation emerges in the learned latent coordinates. Kim et al. (2021) embed an equation-learner (EQL) network after a conventional neural encoder to extract interpretable expressions, so training the whole model end-to-end under a sparsity penalty prunes it to a compact symbolic form. The shared merit of these decomposition methods is those strategies shrink the effective search space, letting each sub-problem be solved with a shorter, more tractable expression. Their common weakness, however, is that the decomposition is discovered from the data itself, either through neural surrogates, detected graph modularity, or learned latent coordinates, rather than drawing on physical theory as references. Consequently the resulting sub-problem need not correspond to physically meaningful mechanisms. This distinction motivates our approach, which instead derives the decomposition from retrieved published theory so that each sub-hypothesis carries an explicit physical interpretation.

3 Agent-Guided Hypothesis-Driven Framework

The Agent-guided Hypothesis-driven Symbolic Regression framework (AHSR) is a pipeline that mimics this scientific process by coupling the capabilities of an LLM with a DRL-based learning process. In AHSR, an LLM acting as a *hypothesis architect* collaborates with a *DRL-based SR explorer* to discover expressions that physically describe the input trajectory data. The *hypothesis architect* carves a physically meaningful decomposition of the target expression, where the decomposition is grounded in the extracted literature and the statistical patterns of the trajectory data. The decomposition specifies the overall topology of the expression assembly (e.g., potential target could be a desired-gap term plus a free-flow term), potential mathematical tokens contained in each subexpression, and a physical dimension constraints. Then the *DRL-based SR explorer* searches within the constrained space defined by each hypothesis portion and produces expression candidates whose assembly can construct the full governing equation. The design of the *hypothesis architect* is to mimic a human developing a governing equation that may contains multiple expression portions corresponding to interpretable meanings. The benefits include: i) each portion's SR search is guided by meaningful mathematical tokens rather than searching blindly over all tokens; and ii) each portion is short enough to sit within a tractable range of the SR search, and the assembled expression reconstructs the full target. The overall AHSR framework can be split into four stages, as depicted in Fig. 1. In the **Hypothesis Forming** stage, the *hypothesis architect* analyzed trajectory data and retrieved the given literature to propose a template that includes a hypothesis portion decomposition skeleton, unit constraints, and potential variables for each hypothesis portion. The *DRL-based SR explorer* then explore potential expressions corresponding to each proposed hypothesis portion. The per-portion exploration includes the processes of sampling and calibration. The **Per-portion Sampling** stage runs dimensionally constrained *DRL-based SR* over each portion's restricted variables, producing expression candidates scored based on an accuracy reward (normalized-RMSE), with two reasons. Firstly, complexity in the sampling process is bounded by a cap that is always below 15 token as a rule of thumb empirically. As this cap confines the search to a bounded complexity range, another parsimony penalty would be redundant. Secondly, physical meaning in AHSR is enforced by units. Physical meaning is addressed in the **Hypothesis Forming** stage, where the

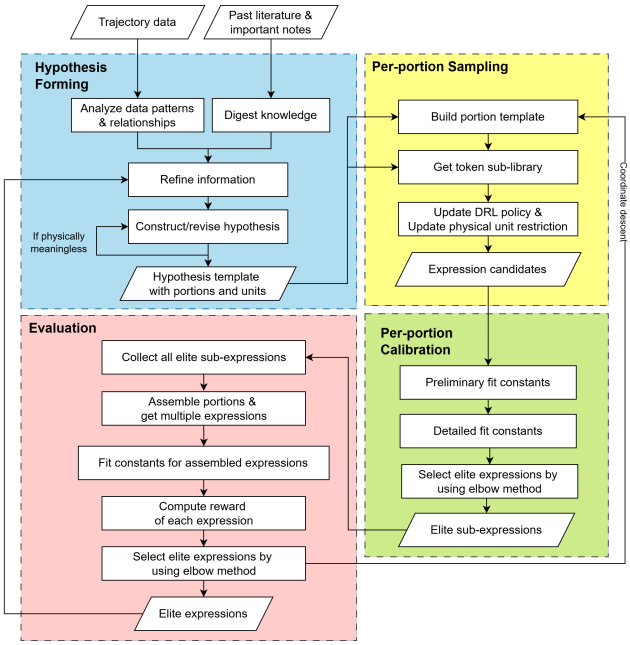

Figure 1: Agent-guided Hypothesis-driven Framework

hypothesis architect revises each portion's variables and a unit-consistency argument. Every candidate produced in the **Per-portion Sampling** stage is already dimensionally valid and physically interpretable, as the DRL policy applies dimensional masking during sampling. The search space itself has been restricted to physically meaningful expressions. The **Per-portion Calibration** stage fits each candidate's free constants and elbow-selects a Pareto set of elite sub-expressions. Then the elite sub-expressions are sent to the **Evaluation** stage and assembled into a full expression. A global constant calibration is then performed for the expression. Afterwards, this stage selects the final elite expressions base on the global calibration accuracy. After global calibration, the updated dependent variable value of each portion goes along the coordinate descent path back to the **Per-portion Calibration** stage while the elite expressions return to the **Hypothesis Forming** stage for the *hypothesis architect* revising the hypothesis template. We describe each stage in the subsections below.

3.1 Problem Formulation

Let $\in \mathbb{R}^{n\times d}$ denote observed driving variables and $\mathbf{y} \in \mathbb{R}^n$ the observed acceleration. The goal is to recover a closed-form model $a = f(\mathbf{X})$ that maximizes the coefficient of determination R^2. HDSR decomposes f into K portions $\{P_k\}_{k=1}^{K}$ combined by an assembly template $\mathcal{T}$:

$$\hat{a} = \mathcal{T}\big(P_0, P_1, \ldots, P_{K-1};\, \mathbf{X};\, \Theta\big),$$

where Θ are global free constants shared across portions. A portion P_k represents a physically interpretable term, such as a free-road acceleration term, an interaction deceleration term, or any sub-expression whose composition under $\mathcal{T}$ reconstructs the full governing function. Each portion P_k is a short, closed-form sub-expression that operates on a restricted subset of the input variables $\mathcal{V}_k \subseteq \{\mathbf{X}\}$ and a small set of free constants $\mathcal{F}_k$;

3.2 Hypothesis Forming

At the start of each iteration, the *hypothesis architect*, an LLM agent returns a complete hypothesis $\mathcal{H} = (\mathcal{T}, \{P_k\}, \ell_k)$, where ℓ_k is the token length cap for this portion after analyzing patterns of the trajectory data, a set of car-following references, and the human suggested K. A hypothesis has three components. First, the hypothesis template $\mathcal{T}$ specifies how the portions combine into a prediction of acceleration, for example, an additive structure $\hat{a} = \alpha\,(P_0 - P_1)$. Second, each portion includes its recommended variable set $\mathcal{V}_k$ and its free constants $\mathcal{F}_k$. Third, a physical unit argument certifies

Algorithm 1 Hypothesis Forming

Input: Data matrix $\mathbf{X}$, target $\mathbf{y}$, retrieval knowledge $\mathcal{C}$, previous hypothesis $\mathcal{H}^{(t-1)}$ (optional), portion count K, stagnation flag *stagnated*
Output: Hypothesis $\mathcal{H} = (\mathcal{T}, \{P_k\}, \ell_k)$

$\mathbf{z} \leftarrow \mathrm{Summarise}(\mathbf{X}, \mathbf{y})$ {ranges, correlations, relationships between variables}
$\mathcal{D} \leftarrow \mathrm{Retrieve}(\mathcal{C}, \mathbf{z})$ {provided car-following references}
$\ell_{\text{hint}} \leftarrow \max\big(8,\ \lfloor \ell_{\max}/K \rfloor\big)$ {per-portion token length cap}
if $\mathcal{H}^{(t-1)} = \varnothing$ **then**
  $\mathcal{H} \leftarrow \mathrm{LLM}_{\text{propose}}(\mathbf{z}, \mathcal{D})$ {initial decomposition and assembly template}
else
  if *stagnated* **then**
    $k^\star \leftarrow \arg\min_k R_k^2$ {weakest portion}
    $\mathcal{H} \leftarrow \mathrm{LLM}_{\text{revise}}(\mathcal{H}^{(t-1)}, k^\star, \mathcal{D})$
  end if
end if
repeat
  if $\mathrm{units}(\mathcal{T}(P_0, \ldots, P_{K-1})) \neq [\mathrm{m/s^2}]$ **then**
    $\mathcal{H} \leftarrow \mathrm{LLM}_{\text{propose}}(\mathbf{z}, \mathcal{D})$
  end if
until unit check passes
return $\mathcal{H}$

that the assembled expression carries units consistent with acceleration, so that dimensionally invalid hypotheses are rejected before any search is run. Note that the number of portions K is given only as a soft suggestion. The *hypothesis architect* is free to choose the decomposition form that the physics and data best support from its opinion. Algorithm1 shows the subroutine in the Hypothesis Forming stage.

3.3 Per-portion Exploration

Each portion P_k is solved independently by a DRL (Guo et al., 2025) model that samples candidate programs as prefix-notation token sequences $\boldsymbol{\tau}$ from a policy π_θ. Instead of searching over all tokens in the library $\mathcal{A}$, the search operates over the restricted token library $\mathcal{A}_k = \mathcal{V}_k \cup \mathcal{F}_k \cup \mathcal{O}$ given by the hypothesis architect for that portion, comprising only its admissible variables $\mathcal{V}_k$, its free constants $\mathcal{F}_k$, and the shared operator set $\mathcal{O}$, where the operator set (e.g., $\{+, -, \times, \div, \ldots\}$) is given at the first place. As the effective search space is $|\mathcal{A}_k| \ll |\mathcal{A}|$ and the token length ℓ_k is capped, the policy explores a space that is orders of magnitude smaller than that of the flat search. During sampling the policy is subject to two hard constraints. First, as in the sampling process in previous works (Petersen et al., 2019; Guo et al., 2025), an arity constraint guarantees that the expression completion is a well-formed expression tree. Second, a physical unit constraint guarantees the constructed expression is dimensionally consistent with the required units of P_k. As a result, every expression the reward evaluates is both well-formed and physically admissible, thus the exploration need not expend capacity discriminating against partially-formed or unit-inconsistent candidates. Policy update follows the risk-seeking method (Petersen et al., 2019), where the gradient is computed only from the top expressions by reward. The reward of portion k is quantified by the standard deviation normalized root-mean-square error,

$$R_k = \left(1 + \frac{\mathrm{RMSE}(\hat{y}_k, y_k)}{\sigma_{y_k}}\right)^{-1} \in (0, 1],$$

where $\hat{y}_k$ is the portion prediction, y_k the derived value of the independent variable, and σ_{y_k} the standard deviation of y_k. Algorithm 2 formalized the subroutine in the **Per-portion Sampling** stage.

3.4 Per-portion Calibration

The **Per-portion Sampling** stage addressed expression structures. What it returns is given a portion P_k, a set of candidate skeletons. The free constants $\mathcal{F}_k$ of each sub-expression candidate have

Algorithm 2 Per-portion Sampling

Input: Portion P_k (variables $\mathcal{V}_k$, constants $\mathcal{F}_k$, horizon ℓ_k), independent variable y_k, policy π_θ, batch size B
Output: Pareto candidate set $\mathcal{P}_k$

$\mathcal{A}_k \leftarrow \mathcal{V}_k \cup \mathcal{F}_k \cup \mathcal{O}$ {restricted token library}
for epoch $e = 1, \ldots, E$ **do**
  $\{\boldsymbol{\tau}^{(b)}\}_{b=1}^{B} \leftarrow \varnothing$
  for $b = 1, \ldots, B$ **do**
    sample prefix sequence $\boldsymbol{\tau}^{(b)}$ from π_θ under length budget ℓ_k, masking tokens that violate dimensional or arity constraints
  end for
  fit free constants of each $\boldsymbol{\tau}^{(b)}$ to y_k {preliminary calibration}
  $R^{(b)} \leftarrow \mathrm{PortionReward}(\boldsymbol{\tau}^{(b)}, y_k)$ {accuracy reward of this portion}
  update π_θ by risk-seeking on the top-quantile rewards
  insert physically valid programs into the Pareto archive $\mathcal{P}_k$
end for
$\mathcal{P}_k \leftarrow \mathrm{ElbowSelect}(\mathcal{P}_k)$ {retain the best candidates}
return $\mathcal{P}_k$

just been preliminary fitted. Before these candidates can be assembled and compared, each skeleton's constants must be well calibrated. The calibration process is the main focus in this stage. We follow the two-step calibration process introduced in the recent work (Guo et al., 2025), where the calibration has two steps including preliminary constant fitting using L-BFGS-B and detailed constant fitting using a hybrid DIviding RECTangles (DIRECT) with sequential quadratic programming (SQP) method (Li et al., 2016). Given the restricted variable set $\mathcal{V}_k$, the preliminary constant fitting is to discard the candidates whose predictions are non-finite or constant. Candidates proceed to detailed constant fitting step if the candidate's fit is good enough, as a skeleton that fits poorly is unlikely to be rescued by another detailed search and would waste compute. The detailed constant fitting step refines the best performed constants $\hat{\mathcal{F}}_k$ and yields the portion's standalone fit $R_k^2 = R^2(\hat{P}_k(\mathbf{X}_{\mathcal{V}_k}), y_k)$. Elbow selection of elite sub-expressions. Candidates then are ordered by their accuracy rewards and selected by the elbow method. The selected candidates are retained as the elite set $\mathcal{P}_k^{\text{elite}}$. This per-portion separation keeps each portion's calibration low-dimensional and conditioned. The best-performed skeletons with approximate constants in the elite set will be globally finalized in the **Evaluation** stage.

3.5 Evaluation and Hypothesis Revise

After every iteration the best portion expressions are assembled, all free constants are jointly re-optimized against the full target by the Nelder–Mead minimization method of the global RMSE. If the assembled R^2 stagnates, the LLM revises the worst-performing portion or the hypothesis structure. The overall loop terminates at a given number of iterations and returns the best-performed assembled expression. The subroutine of the expression assembling and evaluation are formalized in Algorithm 4.

4 Experimental Results

We evaluate AHSR on synthetic dataset. We generate trajectories from two ground-truth car-following models, including the GM model (Chandler et al., 1958) and the IDM (Treiber et al., 2000), under five levels of observation noise. Additive zero-mean Gaussian noise with standard deviation $\sigma \in \{0.0, 0.1, 0.2, 0.4, 0.5\}\,\mathrm{m/s^2}$ is applied to the acceleration signal; the noise-free condition ($\sigma = 0.0$) serves as a clean recovery baseline. Synthetic trajectories are generated with hundred vehicle pairs following the same underlying model. The ground truth expressions are respectively listed in Eq. equation 1 and Eq. equation 2, where in Eq. equation 1 we simplify the sensitivity term c in the GM model as a constant.

Algorithm 3 Per-portion Calibration

Input: Candidate skeleton $\boldsymbol{\tau}$, initial constants $\mathcal{F}_k^{(0)}$, portion target y_k, feature columns $\mathbf{X}_{\mathcal{V}_k}$, loss threshold η (default 0.2)
Output: Calibrated constants $\hat{\mathcal{F}}_k$, portion fit R_k^2

$(\hat{\mathcal{F}}_k,\, \ell_{\text{prelim}}) \leftarrow \mathrm{LBFGS}(\boldsymbol{\tau},\, \mathcal{F}_k^{(0)},\, y_k)$ {preliminary local fitting}
if ℓ_{prelim} is non-finite (NaN) **then**
  discard $\boldsymbol{\tau}$ and **return**
end if
if $\ell_{\text{prelim}} < \eta$ **then**
  $(\hat{\mathcal{F}}_k,\, \ell_{\text{detail}}) \leftarrow \mathrm{DIRECT{+}SQP}(\boldsymbol{\tau},\, \hat{\mathcal{F}}_k,\, y_k)$ {detailed fitting}
else
  retain preliminary constants $\hat{\mathcal{F}}_k$ {skip detailed fitting for poor fits}
end if
$R_k^2 \leftarrow R^2\Big(\hat{P}_k(\mathbf{X}_{\mathcal{V}_k}),\, y_k\Big)$
return $\hat{\mathcal{F}}_k,\; R_k^2$

Algorithm 4 Evaluation

Input: Hypothesis $\mathcal{H}$, Pareto sets $\{\mathcal{P}_k\}$, full target $\mathbf{y}$
Output: Best assembled R^2_{global}, calibrated constants $\Theta^\star$, top assemblies $\mathcal{S}$

$\mathcal{S} \leftarrow \varnothing$
for all combination $(\hat{P}_0, \ldots, \hat{P}_{K-1}) \in \prod_k \mathcal{P}_k$ **do**
  $\hat{a} \leftarrow \mathcal{T}(\hat{P}_0, \ldots, \hat{P}_{K-1})$ {symbolic assembly}
  $r \leftarrow R^2(\hat{a},\, \mathbf{y})$ using portion constants as-is
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{(r,\, \text{combo})\}$
end for
$\mathcal{S} \leftarrow \mathrm{ElbowSelect}(\mathcal{S})$ {retain top assembled candidates}
$(\hat{P}^\star, \ldots) \leftarrow \arg\max_{\mathcal{S}}\, r$
uniquify portion constants; assemble symbolic model g_Θ
$\Theta^\star \leftarrow \arg\min_\Theta \|g_\Theta(\mathbf{X}) - \mathbf{y}\|_2$ {Nelder-Mead joint calibration}
$R^2_{\text{global}} \leftarrow R^2(g_{\Theta^\star}(\mathbf{X}),\, \mathbf{y})$
return $R^2_{\text{global}},\; \Theta^\star,\; \mathcal{S}$

$$a_f(t) = c\,[v_l(t) - v_f(t)]. \tag{1}$$

$$\begin{aligned} a_f(t) &= a_{\max}\left[1 - \left(\frac{v_f(t)}{v_{\max}}\right)^{\lambda} - \left(\frac{\widetilde{s}_f(t)}{s_f(t)}\right)^2\right], \\ \widetilde{s}_f(t) &= s_{\min} + v_f(t)t_{\text{react}} + v_f(t)\frac{(v_f(t) - v_l(t))}{2\sqrt{a_{\max}b}}. \end{aligned} \tag{2}$$

For testing, we use $c = 0.5$ for the GM model while limiting the acceleration to a reasonable range from $-8m/s^2$ to $4\ m/s^2$. As for the IDM dataset, we adopt the widely used calibrated parameters (Treiber et al., 2000): desired velocity $v_{max} = 30\ m/s$, headway $t_{\text{react}} = 1.5\ s$, minimum spacing $s_{min} = 2\ m$, maximum acceleration $a_{\max} = 1\ m/s^2$, comfortable deceleration $b = 1.5\ m/s^2$, and acceleration exponent $\lambda = 4$. The same acceleration bounds of $-8\ m/s^2$ to $4\ m/s^2$ are applied. At each simulation time step t, where we use time interval as 0.1 second, the noiseless acceleration produced by the ground-truth model f is corrupted by an independent Gaussian distribution, $\tilde{a}(t) = f(\mathbf{X}) + \varepsilon(t)$ with $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2)$. The noise level is thus the standard deviation σ of this per-step acceleration perturbation. In the following experiments, AHSR is configured with a maximum of ten hypothesis iterations; the LLM hypothesis architect is GPT-5-class; the literature contains three car-following source paper summaries, including the GM model (Chandler et al., 1958), the Gazis–Herman–Rothery nonlinear follow-the-leader model (Gazis et al., 1961), and IDM

Table 1: GM Model Recovery

Noise Level (σ)	Expression Discovered	R^2
0.0	$a_f(t) = 0.5\,[v_l(t) - v_f(t)]$	1.0
0.1	$a_f(t) = 0.5\,[v_l(t) - v_f(t)]$	0.9
0.2	$a_f(t) = 0.5\,[v_l(t) - v_f(t)]$	0.8
0.4	$a_f(t) = 0.5\,[v_l(t) - v_f(t)]$	0.5
0.5	$a_f(t) = 0.4\,[v_l(t) - v_f(t)]$	0.4

(Treiber et al., 2000). For each reference the retrievable notes encode the acceleration law and its decomposition into physically interpretable terms, the role and admissible units of each input variable, free constants , their units, and the regime each term governs (free-road acceleration, interaction deceleration, desired-gap formation, etc.). This is the structural prior the hypothesis architect draws on when proposing a decomposition, for example, for the GM synthetic data it may retrieve the pure relative-speed law, whereas for richer dynamics it may retrieve the IDM additive free-road/interaction split, depending on the initial data analysis and the exploration performance. All reported R^2 values are computed on data using a 70/30 train-test split. On noise-free synthetic GM trajectories, AHSR recovers an expression structurally equivalent to the GM model, with a $R^2 = 1.0$. The hypothesis architect selects the relative-speed term because the data analysis reports a near-perfect linear association between acceleration and relative speed ($\mathrm{corr}(a_f, \Delta v = [v_f - v_l]) \approx -1$), while spacing and speed are individually almost uncorrelated with the acceleration. Further, the retrieved GM literature supplies the matching relationship. On this evidence the hypothesis architect keeps $[v_f - v_l]$ and settles on the single-term relative-speed response as the most parsimonious structure to the DRL-SR explorer. As noise increases, the recovered skeleton remains the linear relative-speed term, with c declining from ≈ 0.50 at the noise level $\sigma = 0.0$ to ≈ 0.44 at $\sigma = 0.5$. The fit falls as the noise level of the acceleration signal drops. At every noise level the hypothesis architect, after temporarily exploring IDM-style decompositions, converges back to the pure relative-speed GM structure because $[v_f - v_l]$ remains the only informative predictor. The correct qualitative interaction structure is preserved throughout the noise levels. Table 1 summarizes test-set R^2 across noise levels on the GM model. In the IDM experiment, as documented in Table 2, AHSR recovers the model at the noise-free case. From the data analysis report, the hypothesis architect received that acceleration correlates moderately with relative speed, moderately with spacing, and weakly with speed alone. Guided by this report, the hypothesis architect rejects the structure from GM. It is free to propose structures from either GHR, IDM, any combination from those two, or any innovative structure. The LLM revises the hypothesis and proposes an additive IDM decomposition that includes a free-road acceleration term and an interaction deceleration, and this hypothesis guides the explorer to a correct discovery. As noise increases, the exact IDM structure is no longer recovered completely. The discovered expressions still converged to the two-term structure with a free-road term minus an interaction term built from spacing and relative speed. Although the attainable fit remains high, with test $R^2 \approx 0.9$ at both $\sigma = 0.1, 0.2$. At $\sigma = 0.2$ the interaction term is absorbed into a square-root form that sacrifices exact structure to noise, yet still retains spacing and relative speed in the term.

5 Discussion and Conclusion

AHSR displays several advantages on both experiments. It shows its strength to recover both the GM model and IDM. At all the noise levels, this framework recovers the GM model and the harder IDM structurally correct. As the acceleration signal is increasingly polluted, the performance of GM discovery remains the same while IDM discovery decreases. For the cases with noise, although the the LLM ends up with the hypothesis that includes a free-road term and an interaction term, the resulted structures of both lose the correct ratio forms. This indicates the performance ceiling in the high-noise cases is primarily depending on data quality. Although IDM appears in the literature supplied to the model, its governing equation is never provided as ground truth within the reward function. The discovery task is therefore challenging, requiring AHSR to recover the correct structure from data with noise rather than from a hard-coded target. Still, each component in AHSR contributes complementarily, where LLM retrieval improves the quality of the hypothesis proposal, the coordinate-descent update sharpens per-portion targets as the estimate of the portions improves,

Table 2: Discovery on IDM Model

σ	Expression Discovered	R^2
0.0	$a_{\max}\left[1-\left(\frac{v_f(t)}{v_{\max}}\right)^{\lambda}-\left(\frac{\widetilde{s}_f(t)}{s_f(t)}\right)^2\right]$, $\widetilde{s}_f(t)=s_{\min}+v_f(t)\,t_{\text{react}}+v_f(t)\frac{v_f(t)-v_l(t)}{2\sqrt{a_{\max}b}}$	1.0
0.1	$a_{\max}(\lambda-2)+\frac{v_{\max}}{v_f(t)}$ $-a_{\max}\left(\frac{1}{s_f(t)}\left[\frac{b\,(v_f(t)-v_l(t))\,s_f(t)}{s_{\min}\,v_f(t)}+b\right]\right)^2$	0.9
0.2	$k_1\,v_f(t)\,(k_2-1)-a_{\max}+$ $\left(\frac{a_{\max}\,(a_{\max}\,s_f(t)-(v_f(t)-v_l(t))\,k_3\,v_f(t))}{s_f(t)}\right)^{0.5}$	0.9

and the revision loop provides the discrete jumps needed to escape local optima. AHSR also carries a few limitations, some already visible in the completed IDM experiment. The first is a dependence on the quality of the LLM-proposed decomposition. The method inherits the biases of the LLM and its retrieval knowledge, thus a novel expression outside the knowledge may require more iterations for converging. The discovery can converge to a satisfaction soon if the LLM direct the exploration correctly. Once the hypothesis misleads the exploration, on the other hand, it could cause a waste on exploration. Secondly, although the per-portion searches are individually cheap, the end-to-end wall-clock time is dominated by LLM revision iterations. Thirdly, this work could test AHSR on more cases from existing models to actual trajectories. This paper aims to automate the hardest step of model building, the derivation of a compact, physically meaningful functional form. The introduced AHSR is a framework that mimics the way a human scientist actually discovers a governing equation. Instead of searching blindly over tokens, AHSR reproduces the scientific process itself. The framework surveys prior knowledge, understands the observed dynamics, proposes a structured hypothesis, tests it against data, and revises the hypothesis until the expression candidate converges. An LLM acting as a hypothesis architect, informed in the published literature, proposes a physically meaningful decomposition of the target function into short, dimensionally consistent portions together with an assembly template. A physical unit aware DRL-SR explorer then searches each portion independently over a restricted token set, and the portions are reassembled and jointly calibrated, with a coordinate-descent path and an LLM-driven revision loop closing the discovery cycle. By framing symbolic model discovery as human-like and hypothesis-driven loop, AHSR replaces one flat search with several short, physically interpretable sub-searches. This mirrors how a scientist decomposes a complex law into interpretable mechanisms and reasons about each in turn, and it brings targets as complex as a full multi-term car-following model within the tractable task of contemporary SR explorer. Because physical unit is restricted by dimensional masking and expression complexity is bounded by a per-portion token budget, the search reward reduces to a single accuracy evaluation that evaluates every candidate on the same scale and provides an informative gradient while still targeting parsimony, interpretability, generalizability jointly with accuracy. Across our experiments AHSR recovers structures mathematically equivalent to established models and does so by reconstructing the same interpretable components an expert would interpret. The capability of discovering the correct model decreases under noise while the skeletons are mostly retained. These results indicate that the hypothesis-driven decomposition does not trade expression structure for making the search easier but instead recovers the expression an expert would derive by hand. This research work nonetheless has limitations that can be improved. As discussed in the previous section, the framework's success relies on the LLM proposing or refining a hypothesis that settles on a template pointing the search in the right direction. Thus it inherits the biases of the knowledge of the language model. In other words, the initial decomposition will be biased toward structures that are included in the literature, and a novel forms may require more revision iterations before a suitable hypothesis is found. Secondly, the IDM experiment tests only a few cases and this framework has not been tested on unknown real trajectories, which we plan to explore in future research. For this, AHSR leverages the strengths of LLMs to mimic how experts derive governing equations and offers a general pipeline that may accelerate scientific discovery from established knowledge.

ACKNOWLEDGMENTS

We acknowledge the developers and maintainers of the open-source scientific Python packages, including NumPy, SciPy, PyTorch, and the deep symbolic regression frameworks developed by (Petersen et al., 2019) and (Guo et al., 2025). The model in this study is built on the frameworks with the packages. Computational resources were provided by the intelligent Urban Transportation Systems (iUTS) Lab. Finally, we thank OpenAI for API access to the large language models used in this study. The authors declare no competing interests.

USE OF LARGE LANGUAGE MODELS

In this work, we used generative AI tools to propose and refine candidate hypotheses within our pipeline. We did not use generative AI tools to develop theoretical models, formulate or prove mathematical claims, or design the overall research methodology and experimental protocol. Additionally, we used generative AI tools to polish this manuscript for readability.

REFERENCES

Ohay Angah, James Enouen, Xuegang (Jeff) Ban, and Yan Liu. Discovering car-following dynamics from trajectory data through deep learning, 2024. URL https://arxiv.org/abs/2408.00251.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pp. 936–945. Pmlr, 2021.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024. doi: 10.1038/s42256-024-00832-8. Preprint arXiv:2304.05376, 2023.

Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019.

Robert E Chandler, Robert Herman, and Elliott W Montroll. Traffic dynamics: studies in car following. *Operations research*, 6(2):165–184, 1958.

Sam Cox, Andres M. Bran, and Philippe Schwaller. Symbolic regression with large language models. *arXiv preprint arXiv:2402.03177*, 2024.

Miles D. Cranmer, Alvaro Sanchez-Gonzalez, Peter W. Battaglia, Rui Xu, Kyle Cranmer, David N. Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *CoRR*, abs/2006.11287, 2020. URL https://arxiv.org/abs/2006.11287.

Denos C Gazis, Robert Herman, and Renfrey B Potts. Car-following theory of steady-state traffic flow. *Operations research*, 7(4):499–505, 1959.

Denos C Gazis, Robert Herman, and Richard W Rothery. Nonlinear follow-the-leader models of traffic flow. *Operations research*, 9(4):545–567, 1961.

Derek Gloudemans, Yanbing Wang, Junyi Ji, Gergely Zachar, William Barbour, Eric Hall, Meredith Cebelak, Lee Smith, and Daniel B Work. I-24 motion: An instrument for freeway traffic science. *Transportation Research Part C: Emerging Technologies*, 155:104311, 2023.

Zelin Guo, Siqi Wang, Yonglin Tian, Jing Yang, Hui Yu, Xiaoxiang Na, Levente Kovács, Li Li, Petros A. Ioannou, and Fei-Yue Wang. Sr-llm: An incremental symbolic regression framework driven by llm-based retrieval-augmented generation. *Proceedings of the National Academy of Sciences*, 2025. doi: 10.1073/pnas.2516995122.

Samuel Kim, Peter Y. Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4166–4177, 2021. doi: 10.1109/TNNLS.2020.3017010.

John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2):87–112, 1994.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. *arXiv preprint arXiv:2107.14351*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9459–9474, 2020.

Li Li, Xiqun (Michael) Chen, and Lei Zhang. A global optimization algorithm for trajectory data based car-following model calibration. *Transportation Research Part C: Emerging Technologies*, 68:311–332, 2016.

Fei Liu, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang. Algorithm evolution using large language model, 2024.

T. Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P. Santiago, Daniel M. Faissol, and Brenden K. Petersen. Symbolic regression via deep reinforcement learning enhanced genetic programming seeding. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24912–24923, 2021.

Brenden K Petersen, Mikel Landajuela Larma, T Nathan Mundhenk, Claudio P Santiago, Soo K Kim, and Joanne T Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871*, 2019.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. Transformer-based planning for symbolic regression. *Advances in Neural Information Processing Systems*, 36, 2024.

Wassim Tenachi, Rodrigo Ibata, and Foivos I Diakogiannis. Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *The Astrophysical Journal*, 959(2):99, 2023.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, 33:4860–4871, 2020a.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 4860–4871, 2020b.

Lin Xiao, Meng Wang, Wouter Schakel, and Bart van Arem. Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks. *Transportation research part C: emerging technologies*, 96:380–397, 2018.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Zijia Zhong, Earl E Lee, Mark Nejad, and Joyoung Lee. Influence of cav clustering strategies on mixed traffic flow characteristics: An analysis of vehicle trajectory data. *Transportation Research Part C: Emerging Technologies*, 115:102611, 2020.