machines

MDPI

Article

AI-Assisted Bayesian Optimization of a Permanent Magnet Synchronous Motor for E-Bike Applications

Mohammed Abdeldjabar Guesmia [1], Chuan Pham [1], Ya-Jun Pan [2], Kim Khoa Nguyen [1], Kamal Al-Haddad [1] and Qingsong Wang [1,*]

[1] Department of Electrical Engineering, Ecole de Technologie Superieure (ETS), Montreal, QC H3C 1K3, Canada; mohammed-abdeldjabar.guesmia.1@ens.etsmtl.ca (M.A.G.); chuan.pham@etsmtl.ca (C.P.)
[2] Department of Mechanical Engineering, Dalhousie University, Halifax, NS B3H 4R2, Canada; yajun.pan@dal.ca
* Correspondence: qingsong.wang@etsmtl.ca

Abstract

This paper presents an artificial intelligence (AI)-assisted multi-objective topology optimization of a 48 V interior permanent magnet synchronous motor (PMSM) intended for mid-drive e-bike applications. The machine features a 48-slot, 8-pole stator–rotor combination with Δ-shaped three buried magnets per pole, and is coupled to a multi-stage gearbox that adapts its high-speed, low-torque output to a human-scale crank speed. The design problem simultaneously maximizes average torque and efficiency while minimizing torque ripple by varying key stator slot dimensions and magnet geometries. A modular MATLAB–ANSYS Maxwell framework is developed in which finite element simulations are driven by a Bayesian optimization (BO) loop augmented by a large language model (LLM) with retrieval-augmented generation (RAG). The LLM acts as a memory-based agent that proposes candidates, shapes Gaussian Process priors, and incorporates natural language rules expressing qualitative design knowledge. Two AI-assisted trials are compared against a multi-objective Artificial Hummingbird Algorithm benchmark, RAG + BO with and without natural language input. All three methods converge to a similar Pareto region with average torque around 5.4–5.7 Nm, torque ripple of approximately 12.8–14.2%, and efficiency near 93.3–93.6%, suitable for geared e-bike drives. The LLM-guided trial achieves this performance with a 20.1% reduction in simulation expenses relative to the BO baseline and by about 48% compared to the Artificial Hummingbird Algorithm. The results demonstrate that integrating LLM guidance into Bayesian optimization improves sample efficiency while providing interpretable design trends for PMSM topologies tailored for light electric vehicles.

Keywords: bayesian optimization; finite element modeling; large language model; multi-objective optimization; permanent magnet synchronous machine; retrieval-augmented generation

Academic Editor: Yanlei Yu

Received: 1 January 2026
Revised: 22 January 2026
Accepted: 27 January 2026
Published: 1 February 2026

1. Introduction

The increasing demand for energy-efficient and environmentally friendly transportation has accelerated research into advanced electric propulsion systems, with permanent magnet synchronous motors emerging as a key technology in light electric vehicles such as e-bikes and scooters [1–3]. PMSM motors offer superior efficiency, compactness, high

torque-to-weight ratio, and reduced maintenance compared to traditional brushed machines and synchronous reluctance machines [4–9], making them ideal for sustainable mobility applications. Among the inherited drawbacks of a PMSM is the torque ripple elevated by the winding type, slot/pole combination, and design choices [4–6,10,11]. Accurate prediction of motor performance parameters such as back-EMF, torque constant, and inductance often relies on finite element method (FEM) analysis, which has become a standard tool for electromagnetic field evaluation in permanent magnet machines [11,12].

Given the wide range of strategies in which an electrical propulsion system designing process is approached, optimization frameworks and algorithms have been wildly used in electric machine design, parameter identification [13–17], topology [4,18], and control optimization. Multi-objective formulations of FEM-based topology optimizations have proven essential in capturing trade-offs [13,18,19], maximizing average torque and efficiency while minimizing torque ripple. FEM evaluation provides accurate insights into flux distributions, torque ripple, and losses, yet each evaluation is time consuming compared to analytical models, which are less accurate in return. This significantly reduces the number of FEM evaluations needed for the optimization algorithms to converge [4,18,20].

In recent years, artificial intelligence (AI) has emerged as a transformative enabler in electric motor optimization. AI-assisted methods combine surrogate models or machine learning with evolutionary or Bayesian optimization to accelerate FEM-based design optimization [20,21]. Unlike purely heuristic algorithms, AI-driven frameworks can learn correlations from prior simulations to predict performance trends and guide exploration toward promising design regions [20,21]. This reduces computational expenses and enables the discovery of novel topologies that may not arise through conventional search processes.

More recently, large language models (LLMs) have been increasingly investigated as decision-making components for experimental design and Bayesian optimization, providing a natural language interface for encoding expert knowledge, guiding exploration policies, and improving sample efficiency. Recent work has evaluated the practical effectiveness and limitations of LLM-driven experimental design in scientific domains [22]. In parallel, LLM-guided Bayesian optimization frameworks have been proposed for complex multi-objective optimization [23], and reasoning-enhanced Bayesian optimization strategies have been introduced to better exploit long-context inference and iterative hypothesis refinement [24]. Additional efforts have focused on improving the reliability and safety of LLM-in-the-loop optimization through trustworthy BO formulations [25], incorporating interactive natural language feedback from users within the optimization loop [26], and integrating language-guided priors in Bayesian optimization to formalize domain preferences and constraints [27]. Moreover, agentic retrieval-augmented assistants have recently been introduced to translate natural language optimization goals into executable Bayesian optimization pipelines and reusable code templates [28].

This work presents a topology optimization study of a 48 V PMSM motor tailored for e-bike applications. Both rotor and stator geometries are optimized using an AI-assisted Bayesian optimization framework coupled with FEM analysis. The AI module identifies performance trends across the design space and generates refined candidate solutions for Bayesian optimization using retrieval-augmented generation. It also benefits from user input in natural language format using large language models to change user input into optimization policies, which enables efficient searching process. The optimization goal is to simultaneously maximize average torque and efficiency while minimizing torque ripple, yielding a Pareto set of optimal operating points.

To implement this optimization, a framework is created to couple ANSYS Maxwell (2022R2) to MATLAB (2023b) [29]; this allows the interchangeability of optimization algorithms applied to the FEM model and still support any parametric FEM model provided.

To assess the effectiveness of the proposed AI-assisted Bayesian optimization, its performance is evaluated against a multi-objective version of the Artificial Hummingbird Algorithm (MOAHA), selected as a benchmark due to its demonstrated suitability for electric machine topology optimization [19,30]. In the recent literature, MOAHA (and its multi-objective variants) has been successfully applied to the optimization of a PMSM topology closely related to the one investigated in this work [30], and it was systematically compared to established metaheuristics such as PSO and NSGA-II, showing strong convergence and competitive Pareto performance specifically for this class of machines [30]. Therefore, adopting MOAHA as the reference optimizer provides a meaningful and literature-supported baseline for benchmarking the efficiency and solution quality of the proposed approach.

This investigates whether large language model (LLM) guidance can improve the sample efficiency of multi-objective topology optimization for PMSMs. Specifically, how effectively an AI-assisted Bayesian optimization framework coupled with FEM can optimize rotor and stator geometric parameters, and whether retrieval-augmented LLM guidance reduces the number of FEM evaluations required to reach Pareto-optimal solutions compared to a benchmark optimizer previously applied to a closely related problem.

The integration of AI with multi-objective optimization contributes a scalable and intelligent framework for electric motor design, supporting the development of high-performance, energy-efficient electric propulsion systems.

2. Methodology

2.1. Motor Configuration and Design Variables

The e-bike assistance powertrain is a mid-drive system, in which the electric motor transmits its mechanical output to the rider's pedal interface through a built-in gearbox, which serves as a critical mechanism for torque adaptation and rotational speed conditioning (Figure 1).

Figure 1. E-bike assistance powertrain.

The electric motor is a permanent magnet synchronous machine that operates at high rotational velocities and low torque relative to the user pedaling cadence and the torque they need for assistance. Therefore, the electric motor characteristics that are optimal for electromagnetic efficiency are incompatible with the mechanical demands of the human-scale crank motion. To reconcile this mismatch, the motor's shaft output is routed into the multi-stage gearbox which performs fixed torque amplification while reducing rotational speed to a 100 rpm suitable for human pedaling frequency. This arrangement enables the drivetrain to maintain efficient motor operation across variable load conditions, while also ensuring smooth torque delivery to the crankset and pedal spindle.

The motor has 48 slots, 8 poles, and a Δ-type interior magnet topology, where each pole contains three buried permanent magnet segments arranged in a Δ-shaped pattern (Figure 2). This approach is widely used to enhance the flux concentration and provide mechanical robustness. The 48-slot stator accommodates a distributed three-phase winding to smooth the electromagnetic torque and reduce cogging effects. The machine main dimensions are illustrated in Figure 3.

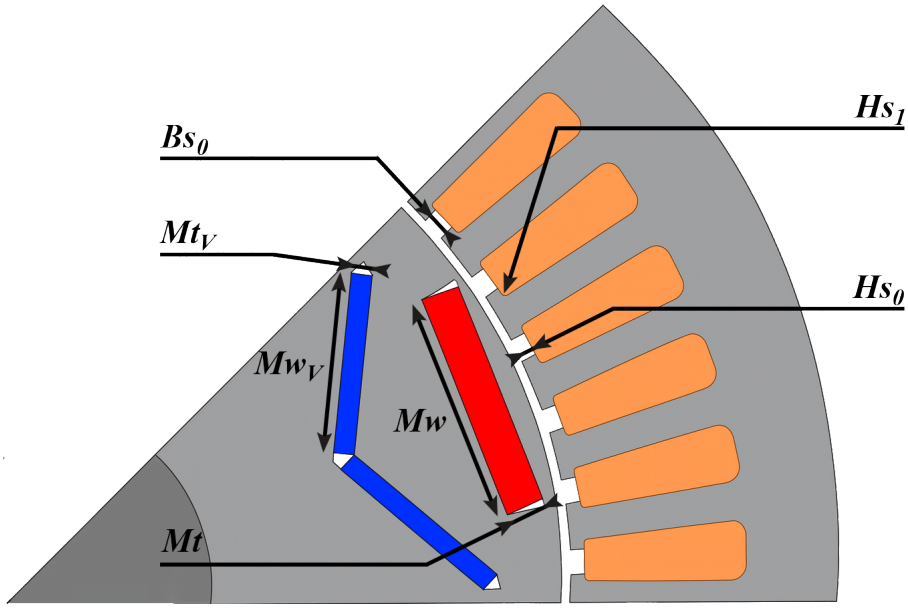


Figure 2. FEM model and parameters.

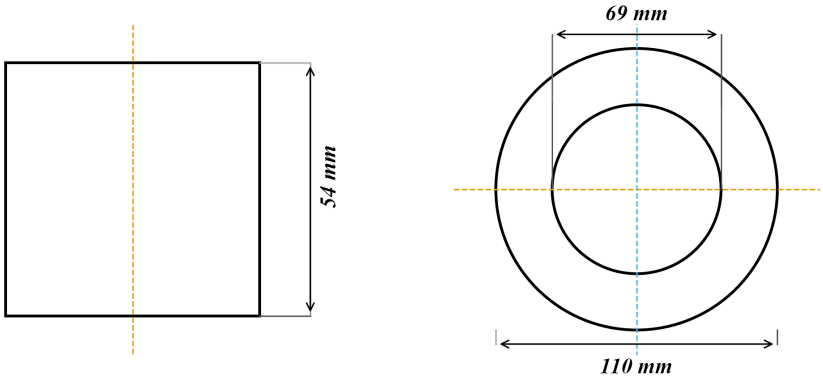


Figure 3. Electric motor main dimensions: (**a**) axial length; (**b**) diameters.

2.2. Finite Element Model

Finite element model: The electromagnetic performance of the machine was evaluated using a 2D finite element model implemented in Ansys Maxwell 2022R2. The model represents a periodic sector of 45° corresponding to one pole (Figure 4). A transient formulation was adopted to compute the motor torque and losses under current excitation, and symmetry was exploited to reduce the simulation time.

Material models and losses: Nonlinear magnetic behavior was considered using B–H curves for M19_24G electrical steel, built-in the ANSYS Maxwell library while magnets were modeled with NdFe35.

The machine efficiency $\bar{\eta}$ is defined as the time-averaged efficiency measured over the last electrical period of an FEM evaluation period. It is computed as the ratio of the mechanical output power P_{mech} to the total converted power. This includes copper losses P_{cu} calculated using Equation (3) where R_{phase} is the phase resistance, and core losses P_{core} that are obtained from the Ansys Maxwell built-in core losses model; it includes hysteresis losses, eddy current losses and excess losses. ${i_A(t)}^2$, ${i_B}^2(t)$ and ${i_C}^2(t)$ are, respectively, phase A, phase B, and phase C instantaneous current.

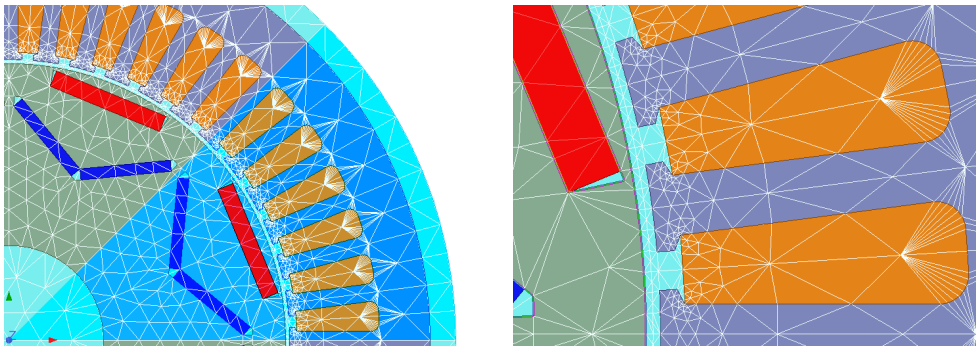
$$\eta(t) = \frac{P_{\text{mech}}(t)}{P_{\text{mech}}(t) + P_{\text{cu}}(t) + P_{\text{core}}(t)} \quad (1)$$

$$\bar{\eta} = \frac{1}{\tau_e} \int_t^{t+\tau_e} \eta(t)\, dt \tag{2}$$

$$P_{cu}(t) = R_{phase} \times (i_A(t)^2 + {i_B}^2(t) + {i_C}^2(t)) \tag{3}$$

The mechanical power is calculated using Equation (4), where $T(t)$ is the instantaneous torque and $\omega(t)$ is the rotor speed.

$$P_{mech}(t) = T(t) \times \omega(t) \tag{4}$$



(**a**) (**b**)

Figure 4. Mesh images: (**a**) global view; (**b**) zoomed-in view.

Mesh strategy and discretization: The domain was discretized using triangular elements (Ansys built-in) with adaptive refinement concentrated in regions with high field gradients, particularly the airgap, magnet edges, and slot teeth tips. The final mesh contains 23,938 elements, with a minimum element area of 1.78×10^{-9} mm^2. Two mesh snapshots are provided in Figure 4a (global view) and Figure 4b (zoom-in of airgap/slots) to illustrate the refinement strategy; Table 1 summarizes the mesh information.

Table 1. Mesh statistics by object.

Object	Num Elements	Min Edge Length (mm)	Max Edge Length (mm)	Min Elem Area (mm^2)	Max Elem Area (mm^2)	Mean Elem Area (mm^2)
Band	1390	0.000500	0.009427	1.38×10^{-7}	3.11×10^{-5}	1.33×10^{-6}
Coil (×1)	98	0.000271	0.003554	5.46×10^{-8}	4.17×10^{-6}	1.17×10^{-6}
V_Magnet_1 (×1)	81	0.000650	0.002724	3.29×10^{-7}	2.11×10^{-6}	6.85×10^{-7}
V_Magnet_2 (×1)	85	0.000650	0.002724	3.29×10^{-7}	2.11×10^{-6}	6.52×10^{-6}
Magnet (×1)	86	0.000772	0.002872	4.86×10^{-7}	2.77×10^{-6}	1.25×10^{-6}
Region	3744	0.000022	0.003000	2.95×10^{-9}	2.58×10^{-6}	1.14×10^{-6}
Rotor	4184	0.000714	0.009459	2.64×10^{-7}	1.30×10^{-5}	2.36×10^{-6}
Stator	7872	0.000022	0.007310	1.78×10^{-9}	1.83×10^{-5}	1.44×10^{-6}
Total	23,938	0.000022	0.009427	1.78×10^{-9}	3.11×10^{-5}	1.25×10^{-6}

2.3. MATLAB–ANSYS Coupling Framework

The proposed optimization environment is structured around a MATLAB–ANSYS coupling that automates the design evaluation loop. MATLAB acts as the supervisory layer, coordinating design space exploration, invoking FE-based simulations, and interpreting results from the iterations. The framework is intentionally modular, allowing for easy replacement of optimization algorithms and FEM models; it also enables integration with Python (version 3.13.5) AI scripts.

As illustrated in Figure 5, the optimization algorithm proposes a new set of design variables and calls MATLAB for an FE evaluation. MATLAB then modifies and launches

dedicated Visual Basic scripts, which in turn update the parametric model in Ansys Maxwell and run the electromagnetic simulation for the current design.

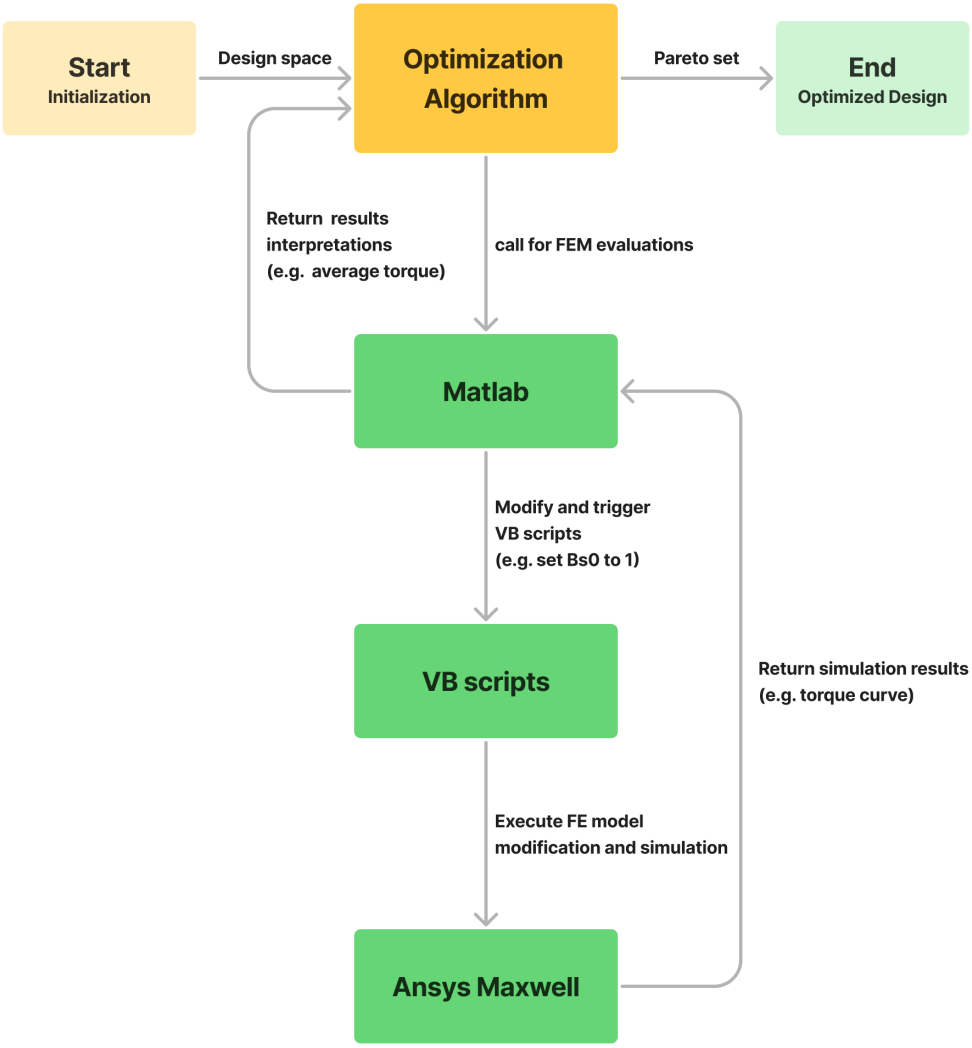


Figure 5. Finite element model optimization framework workflow.

Once the simulation is finished, Ansys Maxwell returns the electromagnetic results (e.g., torque curves) to MATLAB to post-process them into key performance indicators, as average torque, torque ripple, and efficiency, and sends these metrics back to the optimization algorithm. The optimizer uses this feedback to update its search strategy and iterates the loop until a set of optimized motor geometries is obtained.

2.4. Optimization Problem Formulation (Objectives and Constraints)

The PMSM topology optimization problem is formulated as a constrained multi-objective design task, in which the set of the chosen geometric variables is tuned to improve the electromagnetic performance of the e-bike motor. The decision vector is defined as

$$\mathbf{x} = [Bs_0, Hs_0, Hs_1, Mt, Mw, Mt_V, Mw_V]^T, \quad (5)$$

The first set of parameters $\{Bs_0, Hs_0$ and $Hs_1\}$ represents the stator teeth or slot-opening geometry, which contributes to the torque ripple characteristics while the rest of the parameters—Mt, Mw, Mt_V and Mw_V—represent the magnet's dimensions, which affect the amount of torque generated by the machine.

2.4.1. Design Constraints and Feasible Space

The optimization is performed under box constraints defined by practical geometric limits. These bounds establish the feasible design space:

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \tag{6}$$

where the lower and upper values for each parameter are given in Table 2, which directly acts as the constraint definition for the optimization study.

Table 2. Motor design parameters and their corresponding lower and upper bounds used in the optimization study.

Parameter	Lower Bound (mm)	Upper Bound (mm)
Bs_0	0.6	1.8
Hs_0	0.3	0.84
Hs_1	0	0.6
Mt	1.74	3
Mw	4.8	8.4
Mt_V	1.5	1.8
Mw_V	8.4	12.8

These bounds were selected to satisfy mechanical manufacturability and electrical constraints (e.g., minimum slot-opening dimensions, wire gauge considerations, and non-interference constraints during parametric model updates).

2.4.2. Objective Functions

Each candidate design $\mathbf{x}$ is evaluated using the MATLAB–ANSYS Maxwell loop, where FEM simulations return electromagnetic outputs, torque waveform, and losses. These results are post-processed into the key performance indicators used as objectives: average torque, torque ripple, and efficiency. The goal is to identify a Pareto-optimal set of motor geometries that simultaneously perform the following:

- Maximize average torque, $T_{\text{avg}}(\mathbf{x})$;
- Maximize efficiency, $\eta(\mathbf{x})$;
- Minimize torque ripple, $TR(\mathbf{x})$.

This multi-objective formulation can be expressed as Equation (7) and the objective vector as Equation (8), where X is the feasible region defined by the parameter bounds in Table 2. The optimization objective is therefore a three-objective trade-off problem, where improving torque capability conflicts with ripple minimization and efficiency interacts with both.

$$\max_{\mathbf{x}\in X} \left(T_{\text{avg}}(\mathbf{x}),\ \eta(\mathbf{x})\right) \quad \text{and} \quad \min_{\mathbf{x}\in X}\ TR(\mathbf{x}), \tag{7}$$

$$f(x_i) = [T_{\text{avg}}(\mathbf{x}), \eta(\mathbf{x}), TR(\mathbf{x})]^T; \tag{8}$$

Because the performance mapping is obtained through the MATLAB–ANSYS simulation environment, the function is treated as a black-box, providing only objective evaluations without gradient information; therefore, gradient-based optimization is not directly applicable.

2.5. AI-Assisted Bayesian Optimization (RAG + BO + LLM Integration)

The motivation behind this integration is to provide the designer with a prompt where he can request design specification, give experience-based insights, or impose constraints

qualitatively through natural language. The prompted AI agent can then execute actions, assess the results, and refine its policies for the next actions (Figure 6).

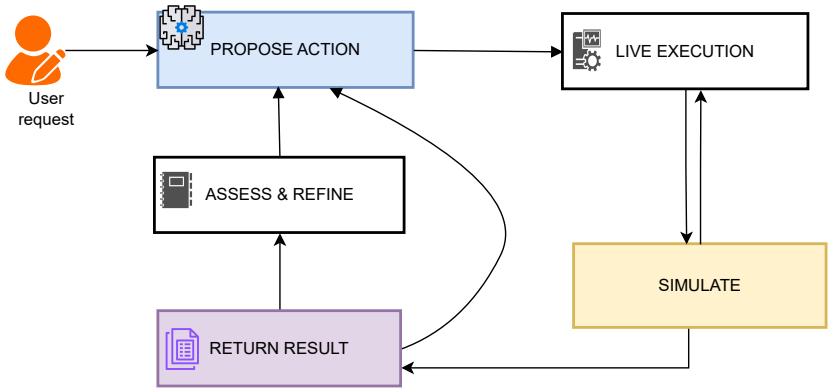


Figure 6. AI agent loop.

BO offers an approximation model to find candidates $x \in \mathcal{X}$. Specifically, an approximation model can be formulated by

$$p(r|x; D_n) = \int_{\Theta} p(r|x, \theta; D_n) p(\theta|x; D_n) d\theta,$$

where $D_n = \{(x_i, r_i)\}_{i=1}^{n}$ is the observed data (the knowledge base), and θ is a latent variable capturing the relationship between p and r. Furthermore, in the standard BO, $p(\theta)$ is often simple (e.g., a wide Gaussian). However, using an LLM with RAG, a useful analysis from humans or history can provide a more informed $p(\theta)$. For example, the LLM might see that increasing Mw can boost torque. Hence, in our model, we define that step by

$$p(\theta|D_n, x) \propto p(D_n|\theta, x) p(\theta),$$

which means how the approximation model incorporates prior knowledge $p(\theta)$, which can be enhanced with LLM insights. Specifically, we use a GP for the approximation method, fitting it to D_n where p_i are parameter sets (e.g., $\{Bs_0 : 1, Hs_0 : 0.5, \ldots\}$) and r_i are the objectives $f(p_i)$ from FEM runs. Based on that, we use the LLM to provide priors $p(\theta)$ or refine predictions. LLM prompts it with D_n to estimate r for a new x, then combines it with GP. For example, $x' = \{Bs_0 : 1.2, Hs_0 : 0.6, \ldots\}$; the GP predicts $\hat{r} = 0.85 \pm 0.1$, and the LLM adjusts based on the current trends (e.g., "wider Bs_0 to reduce ripple") to improve early predictions.

RAG + BO: The high-level overview of the architecture is as follows:

- **Initialization:** The LLM suggests initial parameter sets based on the problem description and human knowledge, e.g., ranges of variable settings. The LLM can make suggestions based on the memory knowledge, which is logged from previous run times. This replaces random initialization for faster convergence.
- **Candiate sampling:** After a few iterations, the LLM generates new candidates to Pareto optimize the cost functions vector $C(x) = [T_{avg}(x), \eta(x), TR(x)]^T$ targeting non-dominated solutions, based on the current knowledge base D_n and a range parameter setting. It uses in-context learning (ICL) with historical data to propose diverse x sets, guided by trends (e.g., increasing Mw boosts torque).
- **Approximation modeling:** The LLM combines with GP to act as an approximation model in finding candidates $x \in \mathcal{X}$ based on the knowledge base D_n and latent variable θ.

- **Evaluation:** This step is run on the black-box simulation MATLAB–ANSYS and it will update the cost value after execution. With the multi-objective function, LLM prompts will trigger the LLM to extract Pareto analysis to process the trade-offs.
- **Stopping:** The algorithm will stop when the Pareto value is stable after many iterations.

Figure 7 illustrates the workflow of the LLM-assisted optimization loop. The agent component mediates between the simulation environment, the knowledge base, and the large language model (LLM). At the start of each iteration, the agent prompts the LLM with the current dataset and past simulation history to suggest promising initial or updated parameter sets and sampling settings. These candidates are passed to the hybrid module of Gaussian Process (GP) and LLM, which uses both a GP surrogate model and the LLM's reasoning to select the next design to evaluate via an acquisition function. The agent then triggers simulations in Ansys, collects the resulting performance metrics, and updates the knowledge base; the LLM analyzes these results relative to previous runs, allowing the system to progressively refine its understanding of the design space and improve subsequent optimization steps.

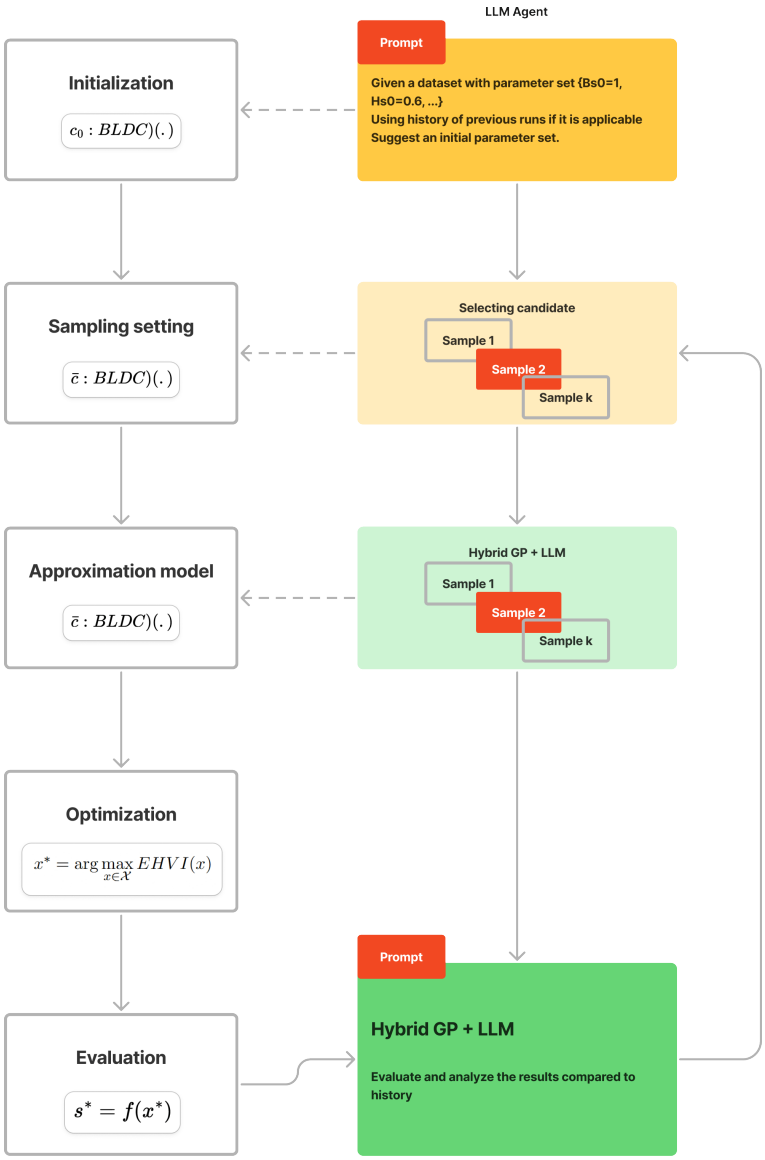


Figure 7. LLM + FEM optimization framework.

In this work, the RAG mechanism is applied in a closed-loop, where retrieval is performed exclusively from the internal knowledge base D_n (historical FEM simulation results) and from LLM-generated memory logs containing trend summaries and qualitative design rules. External unstructured documents (e.g., textbooks or research papers) are not included in the retrieval corpus. While this differs from classic RAG applications that augment LLMs with large external knowledge bases, the use of vector-based semantic retrieval to ground LLM generations in the optimization history aligns with the core principles of retrieval-augmented generation and enables the LLM to serve as an effective, memory-based design agent.

2.6. RAG-BO-LLM-Based Algorithm for Multi-Objective Electric Motor Optimization

Leveraging the significant advantages of the LLM (e.g., a fine-tuned variant of GPT-4 or similar), we build an adaptive agent that injects qualitative design knowledge into the Bayesian optimization (BO) loop, enhancing sample efficiency beyond a standard Gaussian Process (GP) surrogate. However, to address the limitation of pre-trained knowledge in LLMs and preventing hallucinations, we only allow an agent to access the optimization corpus including human interactive guidance, constraints of designs, and prior simulation results (e.g., log files, evaluations). Unlike a traditional RAG system, our solution proposes a dynamic knowledge base as follows:

- **Retrieved information and corpus**: The primary corpus is a dynamic knowledge base $D_n = \{(x_i, r_i)\}_{i=1}^{n}$, where x_i is a vector of design parameters (e.g., $[B_{s0}, H_{s0}, \ldots, M_{wV}]$) and $r_i = [T_{avg}(x_i), \eta(x_i), TR(x_i)]^T$ is the corresponding multi-objective vector from FEM evaluations. This corpus is stored as a vector database for efficient similarity search and updated after each iteration. Retrieval also includes memory logs, such as logs from previous optimization runs, serialized summaries of trends, Pareto analyses, and user-provided natural language rules (e.g., examples in Table 3). Queries to the corpus use semantic similarity (e.g., cosine distance on embeddings generated by the LLM's tokenizer) to fetch the top-k most relevant entries, such as past designs with similar objective trade-offs or parameter ranges. The output of this step is to obtain a list of potential candidates that the LLM will use to make a decision to input into the optimization loop.

Table 3. Natural language input (rules).

Rule ID	Rule Expression
1	Hs_1 has the lowest effect on torque ripple
2	Mt_V has the lowest effect on the average torque
3	The average torque increases by increasing Mw and Mw_V
4	The torque ripple decreases by increasing Bs_0
5	The torque ripple decreases by increasing Hs_0

- **Integration into the optimization loop**: RAG is invoked in key steps (as shown in Figure 7 and Algorithm 1):
  - **Initialization (Lines 7–8 in Algorithm 1)**: The LLM is prompted with the problem description (e.g., optimize the simulation to maximize average torque and efficiency, minimize torque ripple), which is augmented by retrieved memory logs from prior runs (if it has). This generates initial parameter sets, replacing random sampling for faster convergence.
  - **Candidate sampling (Lines 16–18)**: After computing the current non-dominated Pareto set

$$r' = \text{NonDominated}(\{r_i\}_{i=1}^{|D_n|}),$$

the LLM is prompted with D_n (retrieved via similarity to r') and natural language rules (if provided). For example, the prompt is "Based on trends in D_n (e.g., wider B_{s0} reduces ripple) and rules (e.g., increase M_w for higher torque), analyze to generate K = 10 diverse candidates near the Pareto front." With the strong ability of the LLM, it can output candidate sets $\{\tilde{x}_k\}_{k=1}^{K}$, which are filtered for feasibility within bounds X.

- **Approximation modeling (Lines 19–22)**: A standard GP is fit to D_n to predict $\hat{r}_k$ for each $\tilde{x}_k$. RAG enhances this by obtaining trends from D_n (e.g., parameter–objective correlations) and prompting the LLM (e.g., "refine GP prediction for $\tilde{x}_k$ using trends like 'increasing H_{s0} decreases ripple'."). The LLM adjusts the prediction by

$$\hat{r}_k = \alpha_1 \times \mathrm{GP}(\tilde{x}_k) + \alpha_2 \times \mathrm{LLM}(\tilde{x}_k),$$

  where α_1, α_2 are tunable.
- **Evaluation and analysis (Lines 28)**: If trade-offs are detected (e.g., via hypervolume change), RAG retrieves similar past Pareto analyses from memory logs, to prompt the LLM such as "analyze new r^* relative to D_n; suggest policy refinements." This updates memory logs for future retrievals. During the optimization loop, humans can monitor the optimization trend and interrupt to input run-time knowledge in order to reach the optimal solution faster.
- **Stopping**: The algorithm will stop when the Pareto value reaches the threshold.

- **Effects on RAG + BO-based model**:
  - **Priors $p(\theta)$**: To empower the LLM, RAG-injected external but useful knowledge without training for the LLM should be used to help the LLM understand the system and make decisions for optimization such as trends from $\boldsymbol{D_n}$ and rules inform the GP's prior distribution (e.g., biasing the mean function toward regions where wider slots reduce ripple), as in Equation (10): $\boldsymbol{p(\theta|D_n, x) \propto p(D_n|\theta, x)p(\theta)}$, where $\boldsymbol{p(\theta)}$ is shaped by LLM-summarized knowledge, which is dynamic and efficient, to initialize optimization parameters instead of using a randomization method as usual .
  - **Constraints**: Natural language rules act as soft constraints by biasing candidate generation (e.g., penalizing designs violating $\boldsymbol{MtV}$ has low torque effect in prompts), without a hard enforcement in the setting of $\boldsymbol{X}$. This allows for flexible incorporation of finding an optimal design.
  - **Acquisition strategies**: The expected acquisition function (Line 24) is indirectly influenced. Specifically, LLM-generated candidates $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}$ pre-filter the search space, focusing evaluations on promising, knowledge-aligned points. Therefore, this reduces the exploration of dominated regions but it does not change the BO algorithm.

This integration makes the framework interpretable (via natural language prompts/logs) and scalable, as the corpus grows with runs, enabling transfer learning across similar motor designs.

Furthermore, the design architecture implicitly addresses LLM hallucinations or incorrect physical guidance. Specifically, the LLM does not directly control decisions to find the optimal solution. As shown in the architecture, the LLM proposes candidates or adjusts priors, but the final acquisition decision (Line 24: "Select the best candidate using expected improvement") is still driven by the GP-based acquisition function. Hence, even if the LLM suggests a bad setting in an allowed range, the GP will assign it low expected improvement if past data shows poor performance there. Second, our architecture is a RAG-based solution, meaning it conditions its responses on the past simulation results (the

"knowledge base"), possibly engineering settings or rules (e.g., Table 2's natural language rules). This grounds the LLM in observed data, reducing hallucinations. Third, the design of our solution does not rely directly on the LLM. If an LLM-suggested design performs poorly, it is recorded in D_n and down-weighted in future GP predictions. Thus, feedback from real simulation, Ansys, corrects LLM bias over time. Last but not least, the design of α_1 and α_2 limits wrong LLM settings when reaching the optimal solutions. In fact, incorrect physical guidance from the LLM is logged to help the agent avoid those mistakes in future iterations.

Algorithm 1 RAG + BO for Multi-Objective Electric Motor Optimization

Require: Parameter bounds $\mathcal{X}$: Box constraints from Table 2 (e.g., $B_{s0} \in [0.6, 1.8]$ mm, etc.)
Require: Setting: Maximum iterations N, candidates per iteration k, knowledge base D_n,
Ensure: Pareto-optimal parameter set $\mathcal{X}_P$: Non-dominated designs from final D_n

Initialization:

// Prompt LLM with problem description, bounds $\mathcal{X}$, and retrieved memory logs from prior runs
Generate initial parameter sets $\{x_i^0\}_{i=1}^5$ via RAG-augmented LLM
for $i = 1$ to K **do**
  Evaluate x_i^0 using MATLAB–ANSYS FEM simulation
  Compute objectives: $r_i^0 = \left[T_{\text{avg}}(x_i^0), \eta(x_i^0), -TR(x_i^0)\right]^\top$
  Update knowledge base: $D_n \leftarrow D_n \cup \{(x_i^0, r_i^0)\}$
end for
// Main loop
for t = 1 to N **do**
  // Candidate Sampling
  Compute current Pareto reference $r' = \text{NonDominated}(\{r_i\}_{i=1}^{|D_n|})$
  // RAG: Retrieve top-k similar entries from D_n (semantic search on embeddings)
  Prompt LLM with D_n, r', trends (retrieved), and rules: "Generate K diverse candidates near Pareto front"
  Output candidates $\{\tilde{x}_k\}_{k=1}^K$ (ensure feasible in $\mathcal{X}$)
  // Approximation modeling
  Fit independent GPs to each objective in D_n
  // RAG: Retrieve trends from D_n
  Prompt LLM to refine: "Adjust GP mean for $\tilde{x}_k$ based on trends/rules"
  Compute hybrid prediction $\hat{r}_k = \alpha_1 \times \text{GP}(\tilde{x}_k) + \alpha_2 \times \text{LLM-refined mean}(\tilde{x}_k)$
  Select best candidate using Expected Hypervolume Improvement (EHVI)$x^* = \arg\max_k \text{EHVI}(\hat{r}_k | r')$
  Run MATLAB-Ansys FEM for x^* to obtain true $r^* = [T_{avg}(x^*), \eta(x^*), -TR(x^*)]^T$
  Update $D_n = D_n \cup \{(x^*, r^*)\}$
  // RAG: Retrieve similar past analyses;
  Prompt LLM: "Analyze trade-offs in new r^* vs. D_n"
  Update memory logs with LLM summary
  // Stopping criterion
  Measures the volume dominated by the Pareto set P:

$$HV(P) = \text{volume}\left(\bigcup_{r \in P} [r, r_{\text{ref}}]\right)$$

  if $|HV_t - HV_{t-/rho}| / HV_{t-/rho} <$ • over last ρ iterations **then**
    Break loop
  end if
end for
Extract final Pareto front via non-dominated sorting on D_n
return $\mathcal{X}_P = \{x_i \mid r_i \in$ Pareto front$\}$ and best single (e.g., max torque design)

3. Results and Discussion

3.1. AI-Assisted Optimization Without Natural Language Input

The first trial uses the RAG + BO to search for optimal solutions without any natural language input. The results presented in Figure 8 demonstrate a clear convergence pattern toward high-efficiency operating conditions, with the Pareto-optimal solutions forming a compact frontier in the multi-objective design space. In the three-dimensional view (Figure 8a), the Pareto set occupies a narrow region characterized by simultaneously low torque ripple, moderate average torque, and high efficiency.

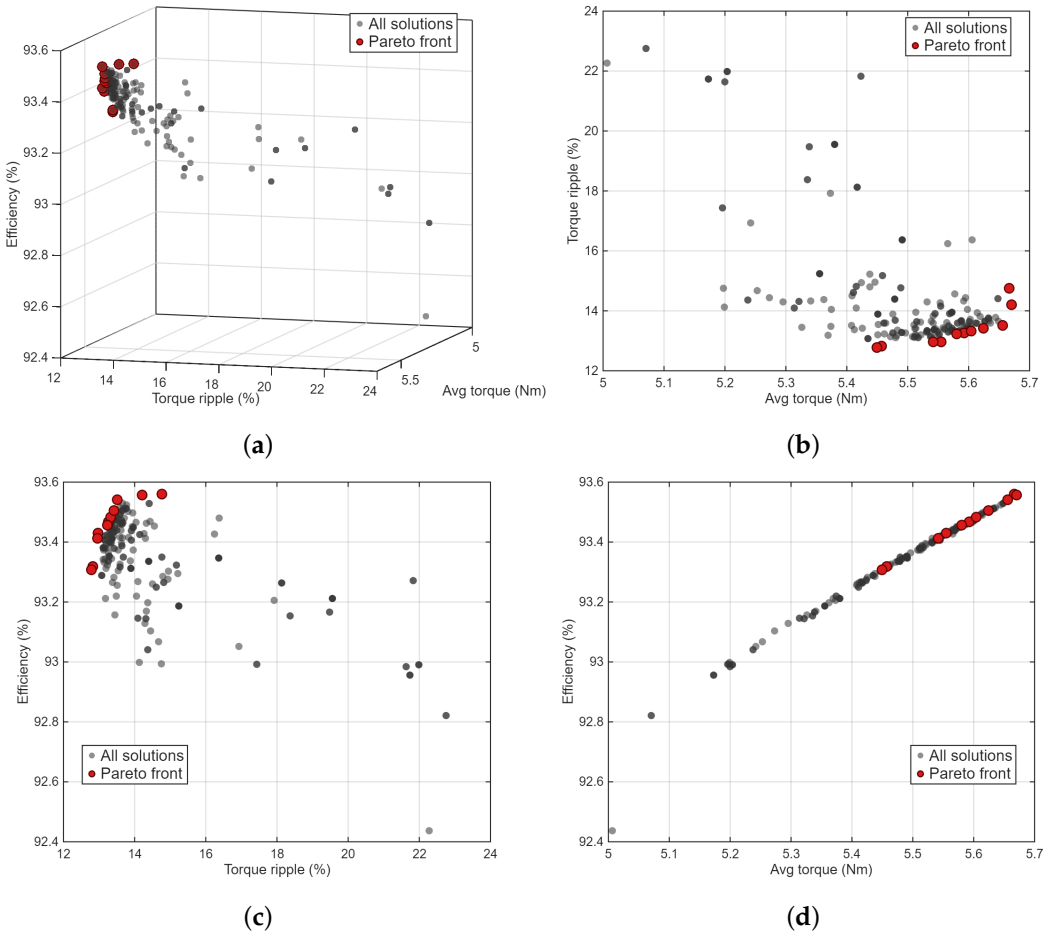


Figure 8. Optimization design points without natural language input: (**a**) 3D view of all solutions and the Pareto front; (**b**) projection, torque ripple vs. average torque; (**c**) projection, efficiency vs. torque ripple; and (**d**) projection, efficiency vs. average torque.

The compact frontier of the Pareto-optimal solutions in the objective space indicates convergence toward a consistent trade-off surface. To verify that this clustering is not an artifact of narrow design variable bounds, we report the explored ranges of each design parameter across all evaluated samples (Table 4), showing that the optimizer sampled broadly within the feasible domain before concentrating near the non-dominated region.

The two-dimensional projections (Figure 8b,c) offer deeper insight into the relationships between the objectives. The torque ripple versus average torque plot (Figure 8b) shows that the Pareto-optimal designs are concentrated toward higher torque–ripple values within the explored region, indicating a trade-off where increased average torque is obtained at the expense of increased torque ripple. Meanwhile, Figure 8c highlights a reversed trend between torque ripple and efficiency, where the Pareto points cluster at the upper-left region, showing that within the explored designs, lower torque–ripple values tend to coincide with higher efficiency.

Table 4. Exploration coverage by variable.

Variable	Min Explored	Min Bound	Max Explored	Max Bound	Range Explored	Bound Range
$Bs0$ (mm)	0.90	0.60	1.80	1.80	0.90	1.20
Hs_0 (mm)	0.44	0.30	0.84	0.84	0.40	0.54
Hs_1 (mm)	0.18	0.00	0.50	0.60	0.32	0.60
Mt (mm)	1.80	1.74	3.00	3.00	1.20	1.26
Mw (mm)	6.02	4.80	8.34	8.40	2.32	3.60
Mt_V (mm)	1.56	1.50	1.80	1.80	0.24	0.30
Mw_V (mm)	10.20	8.40	12.80	12.80	2.60	4.40

Overall, these results demonstrate that the combination of RAG and BO is able to identify feasible design candidates and also to reveal the underlying structure of the performance landscape. The overall design point distribution and the Pareto front knee shape indicate a convergence behavior and suggest that the chosen objective formulations are well-posed for capturing electric motor performance trade-offs.

3.2. AI-Assisted Optimization with Natural Language Input

In the second optimization trial, the AI-assisted Bayesian optimization framework was augmented with natural language input, where qualitative experience-based designs (Table 3) allow the LLM agent to translate design preferences into explicit search policies for the permanent magnet machine. Within the MATLAB–ANSYS coupling, the LLM operates on top of the GP surrogate. In the present trial, this interaction was steered by language instructions that emphasize some of the variables effect on the objectives, for example, which one has the lowest effect on torque, or which one has the highest.

Such a configuration aligns with recent work on LLM-enhanced Bayesian optimization, where natural language priors are used to bias the acquisition function toward user-preferred regions of the design space.

The resulting distribution of design points in the second trial shows a markedly structured exploration of the three-dimensional objective space, similar to the first trial but with fewer explored design configurations. The cloud of gray solutions spans a broad efficiency range, down to approximately ≈92.3%, indicating a sub-optimal start point for the optimization process relative to the previous trial. However, the Pareto-optimal set (red markers) collapses into a compact cluster at simultaneously low torque ripple (≈13–14%) and high efficiency (≈93.4–93.6%), with only modest variations in average torque after only 182 iterations (Table 5). Compared with the first trial, where it took 242 iterations (Table 5), the LLM input resulted in a 20.1% reduction in simulation expenses. This suggests that LLM-guided candidate generation is more effective at sampling along the relevant trade-off surface, rather than wasting evaluations in regions that are sub-optimal in all three objectives.

Table 5. Number of iterations of the first and second trials.

Trial ID	Optimization Technique	Number of Iterations	Number of Candidates
1	RAG + BO without LLM inputs	242	10
2	RAG + BO with LLM inputs	182	10

A closer comparison of the two trials indicates that natural language input reorients the Pareto set toward designs with slightly lower average torque but systematically reduced torque ripple and marginally improved efficiency. In the first trial, the frontier extended

toward designs with higher torque at the cost of increased ripple, consistent with an optimization policy that weighs torque and efficiency more symmetrically against ripple. In contrast, the second trial concentrates Pareto points near the low-ripple edge of the feasible region, indicating that the LLM successfully internalized the designer's qualitative preference and encoded it into the sampling strategy.

From a methodological perspective, the second trial illustrates the principal advantage of embedding an LLM within the optimization loop: the ability to inject high-level engineering knowledge and soft constraints without manually redefining objective weights or hard bounds. The memory augmented, retrieval-based prompting allows the agent to reason over prior FEM evaluations to express this knowledge in new candidate proposals. This knowledge-assisted behavior is consistent with recent studies showing that LLMs can act as adaptive priors or meta-optimizers, improving sample efficiency compared with purely data-driven BO in complex design tasks.

3.3. Comparative Performance Evaluation of the Proposed Method and the MO-AHA

To assess the benefit of the proposed AI-assisted Bayesian optimization, its results are compared with those obtained using the multi-objective version of the AHA. Both optimizers operate on the same MATLAB–ANSYS framework, FEM model, design variables, and bounds defined in Sections 2 and 3, and use the same multi-objective formulation that simultaneously maximizes average torque and efficiency while minimizing torque ripple.

From the 3D scatter plots of the second trial (AI-assisted Bayesian optimization with LLM input; Figure 9) and trial 3 (MOAHA; Figure 10), both methods ultimately identify Pareto-optimal designs located in a similar region in the objective space. This indicates that, in terms of best-found designs, AI-assisted Bayesian optimization is at least competitive with the AHA benchmark, achieving comparable trade-offs between smoothness, efficiency, and torque capability.

However, the search dynamics of the two methods differ. In Trial 3, the MOAHA generates a wide cloud of solutions that spans a wide range of torque ripple (roughly 13–26%) and efficiency (about 92–93.6%), as well as a broad interval of average torque values. The resulting scatter shows a clear global trend from high-ripple and low-efficiency designs toward the low-ripple and high-efficiency region where the Pareto front resides, reflecting large exploration from the meta-heuristic. In contrast, LLM-assisted Bayesian optimization in Trial 2 produces a denser concentration of design points close to the final Pareto cluster, with fewer evaluations spent in clearly dominated regions. This behavior is consistent with the language-encoded rules in Table 2, which bias the search toward the objectives.

The number of FEM evaluations required quantifies the difference in sample efficiency for each trial. LLM-guided Bayesian optimization converged to its compact Pareto set after 182 iterations (Table 6), whereas the Artificial Hummingbird Algorithm required 351 iterations to reach a similar quality of solutions (Table 6). Given the high computational cost of each FEM call, this corresponds to a roughly 48% reduction in simulation effort when using the proposed AI-assisted approach. The improved efficiency arises from exploiting both the surrogate model and the prior knowledge injected through natural language rules and retrieval-augmented memory, which allows the optimizer to focus quickly on the most promising region of the design space.

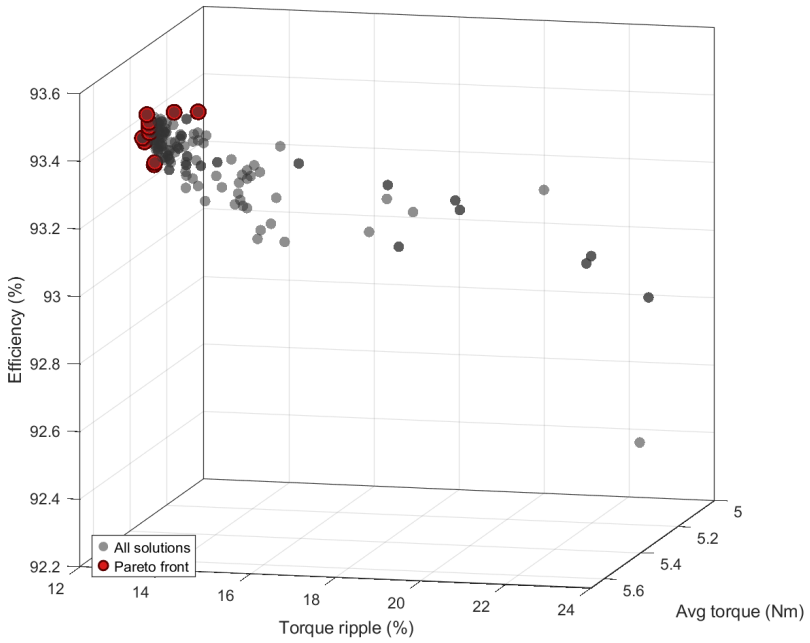


Figure 9. RAG + BO with natural language input results.

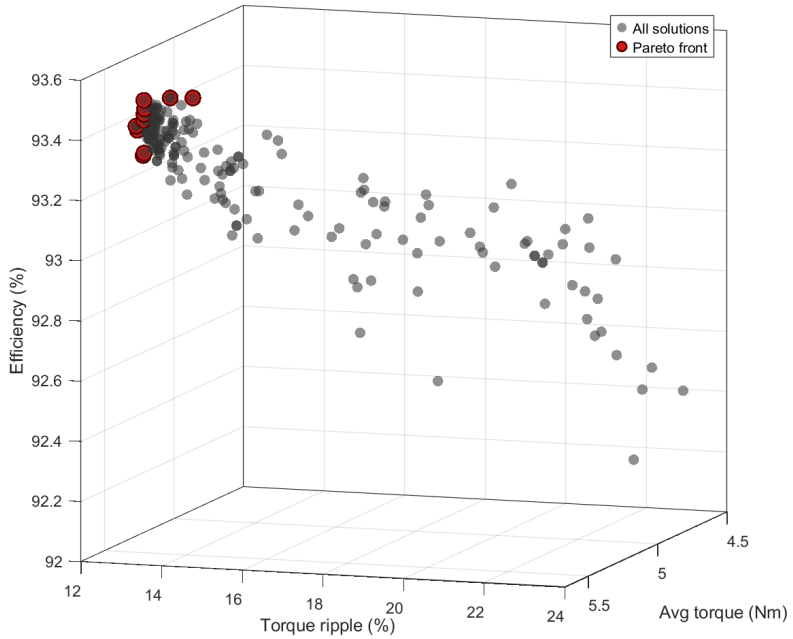


Figure 10. Multi-Objective Artificial Hummingbird optimization results.

Table 6. Number of iterations (second and third trials).

Trial ID	Optimization Technique	Number of Iterations
2	RAG + BO with LLM inputs	182
3	MO-AHA	351

All simulations and optimization runs were executed on a workstation equipped with an Intel Xeon CPU E5-2670 @ 2.60 GHz with eight physical cores (16 threads), 32 GB RAM, running Windows 10. The FEM analysis represents the dominant computational cost

where the average wall-clock time per FEM evaluation was 1 min and 0.62 s (min–max: 59.87 s–1 min 1.87 s) for the mesh settings described in Section 2.

Overall, the comparison shows that the MOAHA provides good global coverage of the search space and serves as a robust benchmark. Nevertheless, AI-assisted Bayesian optimization achieves comparable Pareto performance with substantially fewer simulations and the time reduction is mainly due to the reduction of number of iterations for convergence. For applications such as an electric motor design, where FEM evaluations are the dominate time-consuming aspect of an optimization process, this combination of high-quality solutions and reduced number of evaluations highlights the practical advantage of integrating LLM guidance into the optimization loop.

3.4. Optimal Designs

The optimal design set summarized in Table 7 shows that the RAG + BO framework consistently identifies Pareto-optimal configurations that frequently lie near the upper bounds of several design variables. Most optimal candidates adopt large slot-opening and slot height values ($Bs_0 \approx$ 2.7–3 mm, $Hs_0 \approx$ 1.0–1.4 mm) together with maximum magnet dimensions ($Mt \approx$ 5 mm, $Mw \approx$ 13.9 mm, $Mt_V \approx$ 3 mm, Mw_V close to its upper limit). Across the ten solutions, the average torque remains tightly clustered around 5.5–5.7 Nm while torque ripple and efficiency span a range of ≈12.8–14.2% and ≈93.3–93.6%, respectively, illustrating a well-defined Pareto trade-off. Designs such as Design 1 achieve the highest torque and efficiency at the cost of the largest ripple, whereas designs such as Design 7 minimize ripple but slightly sacrifice torque and efficiency. This behavior is consistent with the multi-objective trends observed in Sections 3.1–3.3, where the Pareto front knee reflects a preference for smoother torque at moderate torque levels suitable for the geared e-bike drivetrain.

Table 7. Optimal motor designs.

Design	Bs_0	Hs_0	Hs_1	Mt	Mw	Mt_V	Mw_V	Average Torque (Nm)	Torque Ripple (%)	Efficiency (%)
1	1.80	0.84	0.42	3.00	8.34	1.80	12.80	5.66	13.51	93.54
2	1.80	0.84	0.42	3.00	8.34	1.80	11.40	5.46	12.83	93.32
3	1.80	0.84	0.42	3.00	8.34	1.80	12.08	5.56	12.97	93.43
4	1.63	0.62	0.38	2.99	8.33	1.78	12.76	5.67	14.21	93.56
5	1.69	0.84	0.37	2.99	8.33	1.75	12.41	5.59	13.26	93.47
6	1.79	0.83	0.41	2.95	8.34	1.75	11.48	5.45	12.78	93.31
7	1.79	0.83	0.38	3.00	8.33	1.72	12.02	5.54	12.96	93.41
8	1.72	0.77	0.39	3.00	8.33	1.73	12.39	5.60	13.37	93.48
9	1.73	0.76	0.40	2.99	8.34	1.80	12.49	5.62	13.42	93.50

Given that the Pareto-optimal candidates exhibit a narrow spread in average torque, the average torque becomes the last discriminator between these designs when manually choosing one of the optimal designs. Efficiency improvements are highly desirable given the current focus on range and power consumption in electrical bikes, provided it does not come at the expense of torque ripple. In this context, Design 9 is the most appropriate choice, since it preserves a near-maximum efficiency (93.50%) while maintaining a comparatively low torque ripple (13.42%) at a representative torque level (5.62 Nm). This selection deliberately avoids the highest-efficiency option (Design 4, 93.56%), which has noticeably larger ripple (14.21%) for a marginal efficiency gain. Design 4 offers the lowest ripple in the set, but it also offers the lowest average torque (5.45 Nm) and a lower efficiency

(93.31%) compared to Design 9. Overall, Design 9 offers the most balanced compromise in the Pareto set.

4. Conclusions

This work proposed and evaluated an AI-assisted optimization framework for the topology design of a Δ-type interior PMSM dedicated to e-bike mid-drive powertrain. By coupling a MATLAB–ANSYS Maxwell simulation environment with a Bayesian optimization scheme enhanced by an LLM-based, retrieval-augmented agent, the study addressed a three-objective problem that seeks to maximize average torque and efficiency while minimizing torque ripple. Across all trials, the optimizer identified feasible geometries that deliver moderate average torque (≈5.3–5.7 Nm), low torque ripple (≈12.8–14.2%), and high efficiency (≈93.3–93.6%), which are compatible with the requirements of a geared e-bike drivetrain.

The comparison between the purely data-driven RAG + BO baseline and the LLM-guided RAG + BO configuration highlights the benefit of incorporating natural language design knowledge directly into the search process. When qualitative rules on the influence of slot and magnet parameters were provided, the LLM reoriented the exploration toward smoother torque, high-efficiency designs and concentrated samples near the relevant trade-off surface. This led to convergence in 182 iterations, representing a 20.1% reduction in simulations relative to the 242 iterations required by the baseline trial without language guidance.

Against the multi-objective Artificial Hummingbird Algorithm benchmark, AI-assisted Bayesian optimization achieved a comparable Pareto front while requiring substantially fewer simulations (182 versus 351 iterations, a reduction of about 48%). MOAHA offered broad global exploration, but the proposed approach exploited both surrogate model information and LLM-encoded priors to avoid clearly dominated regions of the design space. This improved sample efficiency is particularly valuable in electric machine design, where each high-fidelity FEM evaluation is computationally expensive.

The resulting optimal designs also provide physical insight. Most Pareto-optimal solutions push the stator slot-opening, slot height, and magnet dimensions close to their upper bounds, confirming that increased magnet volume and wider slots promote torque production while maintaining acceptable ripple and efficiency. The fact that the best candidates cluster at these limits, without clear signs of magnetic saturation, suggests that relaxing some geometric bounds in future studies could unlock additional torque and efficiency gains.

Author Contributions: Conceptualization, M.A.G.; software, M.A.G. and C.P.; validation, Q.W., Y.-J.P., K.A.-H., and K.K.N.; resources, M.A.G., C.P., Q.W., and K.K.N.; writing—original draft preparation, M.A.G.; writing—review and editing, Q.W., Y.-J.P., and K.A.-H.; visualization, M.A.G. and C.P.; supervision, Q.W. and K.K.N.; project administration, Q.W. and K.K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by The Natural Sciences and Engineering Research Council of Canada (NSERC).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chawrasia, S.K.; Das, A.; Chanda, C.K.; Banerjee, S. Design analysis and comparative study of hub motor for an electric bike. In Proceedings of the Michael Faraday IET International Summit (MFIIS 2020), Kolkata, India, 3–4 October 2020; pp. 1–6.
2. Dash, A.; Kumar, S.; Sahoo, A.K. Design and analysis of electric bike (E-bike). In *Sustainable Expert Systems, Proceedings of the ICSES 2021, Antalya, Turkey, 11–14 November 2021*; Springer: Singapore, 2023; pp. 1–7.
3. Chawrasia, S.K.; Das, A.; Chanda, C.K.; Banerjee, S. Design and analysis of electric bike hub-motor using Motor-CAD. In *Proceedings of the 2021 3rd International Conference on Energy, Power and Environment (ICEPE), Shillong, India, 5–7 March 2021*; IEEE: New York, NY, USA, 2021; pp. 1–6.
4. Duan, Y.; Ionel, D.M. A review of recent developments in electrical machine design optimization methods with a permanent-magnet synchronous motor benchmark study. *IEEE Trans. Ind. Appl.* **2013**, *49*, 1268–1275. [CrossRef]
5. Nasiri-Zarandi, R.; Karami-Shahnani, A.; Toulabi, M.S.; Tessarolo, A. Design and experimental performance assessment of an outer rotor PM-assisted SynRM for the electric bike propulsion. *IEEE Trans. Transp. Electrif.* **2023**, *9*, 727–736. [CrossRef]
6. Wu, Y.; Huang, H.; Zou, T.; Ren, X.; Gerada, C. Design of a 1200 Nm High Torque Density In-Wheel Magnetic Geared Motor for Electrical Vehicles. In Proceedings of the 2024 IEEE Vehicle Power and Propulsion Conference (VPPC), Washington, DC, USA, 7–10 October 2024; pp. 1–6.
7. Ho, P.-J.; Yi, C.-P.; Lin, Y.-J.; Chung, W.-D.; Chou, P.-H.; Yang, S.-C. Motor torque control of electric assist bike considering external resistances. In *Proceedings of the IECON 2023—49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 16–19 October 2023*; IEEE: New York, NY, USA, 2023; pp. 1–7.
8. Ho, P.-J.; Yi, C.-P.; Lin, Y.-J.; Chung, W.-D.; Chou, P.-H.; Yang, S.-C. Torque measurement and control for electric-assisted bike considering different external load conditions. *Sensors* **2023**, *23*, 4657. [CrossRef]
9. Mohanraj, D.; Aruldavid, R.; Verma, R.; Sathyssekar, K.; Barnawi, A.B.; Bharathiraja, C.; Mihet-Popa, L. A review of BLDC motor: State of art, advanced control techniques, and applications. *IEEE Access* **2022**, *10*, 54833–54869. [CrossRef]
10. Krishnan, P.; Marimuthu, N.S.; Vetrichelvan, V.; Karthikeyan, V.; Sangeetha, R. Cogging torque reduction strategies in six-phase BLDC motors: Skewing and back EMF tuning for EV applications. In *Proceedings of the 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 6–8 November 2024*; IEEE: New York, NY, USA, 2024; pp. 1777–1782.
11. Chen, H.; Meng, Y.; Zhang, Q.; Li, D.; Qu, R. Design and Analysis of a New Asymmetric Consequent-Pole Flux Reversal Dual-PM Excited Machine with Trapezoidal PMs. *IEEE Trans. Transp. Electrif.* **2025**, *11*, 8702–8713.
12. Zhao, Y.; Li, D.; Liang, Z.; Qu, R. Design Considerations of Spoke-Array PM Vernier Machine for Different Power Levels. *IEEE Trans. Ind. Appl.* **2025**, *61*, 6149–6159. [CrossRef]
13. Knebl, L.; Barta, J.; Bramerdorfer, G.; Vitek, O.; Ondrusek, C. Multi-objective optimization of a line-start synchronous machine using a self-organizing algorithm. *IEEE Trans. Magn.* **2021**, *57*, 9344813. [CrossRef]
14. Parouha, R.P.; Verma, O.P. State-of-the-art reviews of meta-heuristic algorithms with their novel proposal for unconstrained optimization and applications. *Arch. Comput. Methods Eng.* **2022**, *29*, 2749–2791. [CrossRef]
15. Yang, C.; Liu, W.; Niu, S.; Lyu, J.; Chau, K.T. Parameter-Tuning-Free Two-Step Identification of Mechanical Parameters for PMSM Drives. *IEEE Trans. Ind. Electron.* **2025**, *72*, 12378–12392. [CrossRef]
16. Escarela-Perez, R.; Niewierowicz, T. Synchronous machine parameters from frequency-response finite-element simulations and genetic algorithms. *IEEE Trans. Energy Convers.* **2001**, *16*, 198–203. [CrossRef]
17. Belahcen, A.; Martin, F.; El Hadi Zaim, M.; Dlala, E.; Kolondzovski, Z. Combined FE and particle swarm algorithm for optimization of high speed PM synchronous machine. *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.* **2015**, *34*, 475–484.
18. Zheng, S.; Zhu, X.; Xu, L.; Xiang, Z.; Quan, L. Multi-objective optimization design of a multi-permanent-magnet motor considering magnet characteristic variation effects. *IEEE Trans. Ind. Electron.* **2022**, *69*, 3428–3438.
19. Zhao, W.; Zhang, Z.; Mirjalili, S.; Wang, L.; Khodadadi, N.; Mirjalili, S.M. An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Comput. Methods Appl. Mech. Eng.* **2022**, *398*, 115223.
20. Murthy, P.K.; Mopari, S.S.; Hirekhan, S.R. Optimization of electric motor design using AI-based algorithm. *Nanotechnol. Percept.* **2024**, *20*, 2139–2158.
21. Liu, T.; Astorga, N.; Seedat, N.; van der Schaar, M. Large language models to enhance Bayesian optimization. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
22. Gupta, R.; Hartford, J.; Liu, B. LLMs for Bayesian Optimization in Scientific Domains: Are We There Yet? In *Findings of the Association for Computational Linguistics, Proceedings of the EMNLP 2025, Suzhou, China, 4–9 November 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 15482–15510.
23. Zhang, Y.; Choong, J.J.; Madhawa, K.; Ozawa, K. AutoLead: An LLM-Guided Bayesian Optimization Framework for Multi-Objective Lead Optimization. *bioRxiv* **2025**. [CrossRef]

24. Yang, Z.; Wang, D.; Ge, L.; Wang, B.; Fu, T.; Li, Y. ReasoningBO: Enhancing Bayesian Optimization with the Long Context Reasoning Power of LLMs. *arXiv* **2025**, arXiv:2505.12833.
25. Chang, C.-Y.; Azvar, M.; Okwudire, C.; Al Kontar, R. LLINBO: Trustworthy LLM-in-the-Loop Bayesian Optimization. *arXiv* **2025**, arXiv:2505.14756.
26. Kobalczyk, K.; Lin, Z.J.; Letham, B.; Zhao, Z.; Balandat, M.; Bakshy, E. LILO: Bayesian Optimization with Interactive Natural Language Feedback. *arXiv* **2025**, arXiv:2510.17671. [CrossRef]
27. Topalis, P.; Schieseck, M.; Gehlhoff, F. LangBO: A Framework for Language-Guided Prior Integration in Bayesian Optimization. In Proceedings of the 2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA), Porto, Portugal, 9–12 September 2025. [CrossRef]
28. Sayeed, H.M.; Soneji, A.; Baird, S.; Sparks, T. Honegumi RAG Assistant: An Agentic System for Accelerating Bayesian Optimization Adoption in Experimental Sciences. *ChemRxiv* **2025**. [CrossRef]
29. Guzmán-Romero, D.; Sauceda-Peña, B.; Sánchez-García, S.; Mantilla-Pérez, P. Development of a MATLAB–GAMS framework for solving the problem regarding the optimal location and sizing of PV sources in distribution networks. *Resources* **2023**, *12*, 35.
30. Zhang, S.; Yan, H.; Yang, L.; Zhao, H.; Du, X.; Zhang, J. Optimization Design of Permanent Magnet Synchronous Motor Based on Multi-Objective Artificial Hummingbird Algorithm. *Actuators* **2024**, *13*, 243. [CrossRef]