MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Schema-Guided BERT-LoRA Framework for Continual Surrogate Modeling of Surface Permanent Magnet Motors

Li, Yuxiao; Koike-Akino, Toshiaki; Wang, Ye; Yamamoto, Tatsuya; Sakamoto, Yusuke; Wang, Bingnan

TR2026-157 October 28, 2026

Abstract

Surrogate models can substantially reduce the finiteelement-analysis cost in permanent magnet motor design, but conventional parameter-based models are often tied to a fixed set of input variables. As a result, even small changes in the motor schema may require redesigning the model architecture or rebuilding the data-processing pipeline. This paper presents a schema-guided continual surrogate modeling framework based on BERT and low-rank adaptation (LoRA). Motor parameters are serialized into canonical text descriptions, allowing heterogeneous schemas to be handled by a shared language-model backbone. To support incremental design workflows, a base surrogate is first trained on the original surface permanent magnet motor schema, while newly arriving schema-expanded variants are adapted using lightweight task-specific LoRA modules and regression heads. To avoid task interference caused by sequential LoRA adaptation, each variant uses an isolated adapter selected by a schema registry during inference. Experiments on three related motor datasets show that the proposed framework enables efficient adaptation to schema-modified designs, preserves the performance of previously learned tasks, and remains robust when input parameter clauses are reordered.

IEEE Industrial Electronics Society (IECON) 2026

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Schema-Guided BERT-LoRA Framework for Continual Surrogate Modeling of Surface Permanent Magnet Motors

Yuxiao Li
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115 USA

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA

Ye Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA

Tatsuya Yamamoto
AI Transformation Innovation Center
Mitsubishi Electric Corporation
Kamakura, Kanagawa 247-8501 Japan

Yusuke Sakamoto
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan

Bingnan Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA

***Abstract*—Surrogate models can substantially reduce the finite-element-analysis cost in permanent magnet motor design, but conventional parameter-based models are often tied to a fixed set of input variables. As a result, even small changes in the motor schema may require redesigning the model architecture or rebuilding the data-processing pipeline. This paper presents a schema-guided continual surrogate modeling framework based on BERT and low-rank adaptation (LoRA). Motor parameters are serialized into canonical text descriptions, allowing heterogeneous schemas to be handled by a shared language-model backbone. To support incremental design workflows, a base surrogate is first trained on the original surface permanent magnet motor schema, while newly arriving schema-expanded variants are adapted using lightweight task-specific LoRA modules and regression heads. To avoid task interference caused by sequential LoRA adaptation, each variant uses an isolated adapter selected by a schema registry during inference. Experiments on three related motor datasets show that the proposed framework enables efficient adaptation to schema-modified designs, preserves the performance of previously learned tasks, and remains robust when input parameter clauses are reordered.**

***Index Terms*—Permanent magnet motor, surrogate model, BERT, LoRA, schema canonicalization, continual learning.**

I. Introduction

Permanent magnet motors are core components in appliances, electric transportation, factory automation, and high-efficiency motion-control systems. Surface permanent magnet (SPM) motors are particularly attractive because of their simple rotor structure and high power density. However, their performance is sensitive to rotor geometry, magnet dimensions, tooth-tip shape, slot opening, and air-gap design. Design engineers therefore need to evaluate many candidate geometries before finalizing a motor design.

High-fidelity finite-element analysis (FEA) is widely used for motor performance prediction, but repeated simulations can become computationally expensive in large parameter sweeps and optimization loops. Surrogate modeling reduces this cost by replacing full-order simulations with learned regression models [1]–[5]. Existing surrogate models commonly use numerical design vectors, image-based geometry representations, or magnetic-field distributions [6]–[9]. However, vector-based models are typically tied to a fixed parameter set, while image- and field-based representations often require more involved preprocessing.

This limitation becomes particularly relevant during iterative motor design. A surrogate may initially be trained for one motor schema, while subsequent design refinements introduce additional features such as dummy slots or magnet stoppers. Although these variants remain closely related to the original motor, their expanded parameter schemas are no longer directly compatible with a fixed-dimensional surrogate. Rebuilding or jointly retraining the model after every schema update can be inefficient, especially when only a limited number of new FEA samples are available for each variant.

A natural alternative is sequential fine-tuning, in which the same surrogate is progressively updated from one schema to the next. However, this strategy can lead to catastrophic forgetting [10], [11]: adapting the model to a new schema may degrade its performance on previously learned variants. This is particularly problematic for quantities such as peak-to-peak cogging torque, which depend on fine geometric interactions among the slot opening, tooth-tip profile, magnet geometry, and air-gap design. Training a separate full model for every variant avoids such interference but is parameter-inefficient and becomes increasingly inconvenient as the number of motor variants grows.

Recent NLP studies have explored LoRA-based strategies for continual, multi-task, and cross-task adaptation, including MoE-style LoRA experts, orthogonal low-rank subspace constraints, and learned composition of LoRA modules trained

This work was done when Y. Li was with MERL as research intern. Corresponding Author: Bingnan Wang (bwang@merl.com).

on different tasks [12]–[15]. These methods manage task differences through learned routing, module composition, or subspace constraints. For motor surrogate modeling under schema evolution, however, the task identity is explicitly determined by the parameter schema. This allows learned routing or module composition to be replaced with deterministic schema matching.

We therefore propose a text-based BERT-LoRA framework for continual surrogate modeling across related SPM motor tasks. Each motor design is serialized into a canonical parameter-value description and processed by a BERT-style encoder [16] , instantiated as DistilBERT [17] in our experiments. A base surrogate is first trained on the original schema, while each newly arriving variant receives its own LoRA adapter [18] and regression head with the shared backbone kept frozen. At inference time, a schema-guided registry extracts field-value pairs, restores the canonical field order, identifies the corresponding motor schema, and activates the appropriate adapter and regression head.

The main contributions are:

- A text-based surrogate formulation for SPM motor performance prediction under evolving parameter schemas, using canonical parameter-value descriptions rather than fixed-dimensional numerical inputs.
- A continual adaptation strategy based on task-specific LoRA adapters, enabling a base surrogate to be extended to schema-expanded motor variants without full backbone retraining or sequential overwriting of shared model weights.
- A schema-guided registry for multi-task deployment that deterministically identifies the target motor schema, selects the corresponding adapter and regression head, and reconstructs the canonical field order for robust inference under shuffled input clauses.

II. Problem Definition

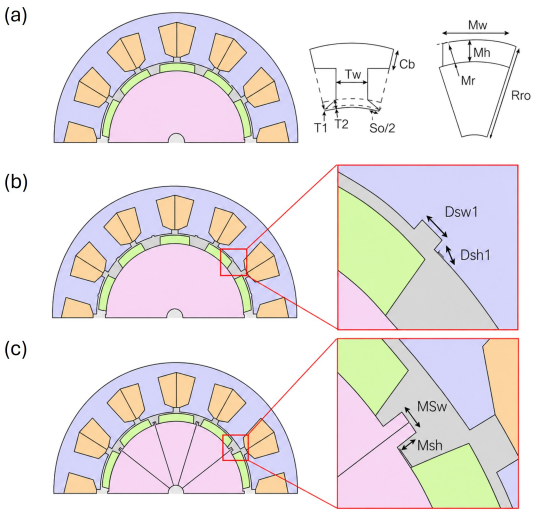

Fig. 1. Three related SPM motor design settings: (a) original design, (b) dummy-slot variant, and (c) magnet-stopper variant.

We study data-efficient surrogate modeling for a family of related SPM motor design tasks with small geometric modifications. As illustrated in Fig. 1, dataset D0 corresponds to the original design and is described by 9 structural parameters. D1 and D2 are schema-expanded variants derived from D0 by introducing a dummy-slot feature and a magnet-stopper feature, respectively. Consequently, the input dimensionality increases from 9 to 11 for both D1 and D2. This setting reflects a practical adaptation scenario in which a surrogate trained for the original design must be efficiently extended to related designs with slightly modified parameter schemas.

TABLE I. Input parameters of the three related motor tasks.

Abbr.	Meaning	D0	D1	D2
Rro	Rotor outer radius	✓	✓	✓
Mh	Magnet height	✓	✓	✓
Mw	Magnet width	✓	✓	✓
Mr	Magnet curve radius	✓	✓	✓
T1	Tooth tip height 1	✓	✓	✓
T2	Tooth tip height 2	✓	✓	✓
Tw	Tooth width	✓	✓	✓
Cb	Core-back thickness	✓	✓	✓
So	Slot opening	✓	✓	✓
Dsw1	Dummy-slot width		✓	
Dsh1	Dummy-slot height		✓	
MSw	Magnet-stopper width			✓
MSh	Magnet-stopper height			✓

For each design, FEA provides the target motor performance. We focus on two scalar quantities shared by all tasks: peak-to-peak cogging torque, denoted by cog_{pp}, and effective back-EMF, denoted by emf_{eff}. For task $t \in \{0, 1, 2\}$, each sample consists of an input parameter set $x_i^{(t)}$ and a target vector $y_i^{(t)} = [\mathrm{cog}_{pp,i}^{(t)}, \mathrm{emf}_{eff,i}^{(t)}]$. The goal is to learn task-dependent predictors $f_t : x^{(t)} \mapsto y^{(t)}$ while exploiting the shared structure among related motor designs.

The three tasks are introduced incrementally, with D0 available first and schema-expanded variants D1 and D2 arriving later. We therefore consider schema-continual surrogate modeling, where a D0-trained base model is extended to new tasks using new data while retaining performance on previously learned schemas. Sequential fine-tuning provides a natural baseline but may introduce cross-task interference, motivating the task-isolated strategy evaluated in Section IV-D1.

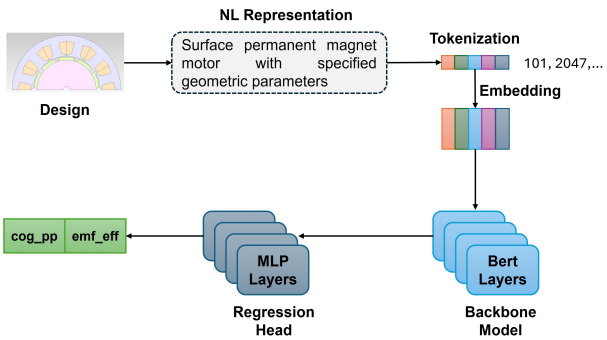

Fig. 2. Text-based surrogate workflow. Motor parameters are serialized into a canonical description, processed by a BERT encoder, and mapped by an MLP head to cog_{pp} and emf_{eff}.

III. Proposed Methodology

A. Text-Based Motor Representation

Instead of feeding raw numerical vectors directly into the model, each sample is converted into a canonical text description. For a task with fields $\{p_i\}_{i=1}^{n_t}$, the description is formed by concatenating parameter-value fragments in the schema order:

$$s^{(t)} = \mathrm{concat}\{\phi(p_1), \ldots, \phi(p_{n_t})\}. \tag{1}$$

Here $\phi(p_i)$ denotes a text fragment that contains a canonical field name and its raw numerical value in the original physical unit. In this work, all geometric parameters are serialized in millimeters.

For example, a D0 sample is serialized as: "This is a surface permanent magnet motor design... rotor outer radius is 24.11; magnet height is 3.92; ...; slot opening is 0.99." Each numerical value is therefore explicitly associated with its corresponding field name and design context.

B. BERT-Style Regression Backbone

Fig. 2 shows the text-based surrogate workflow. The canonical description $s^{(t)}$ is first tokenized and embedded, and the resulting token sequence is passed to a BERT-style text encoder. Let h_{cls} denote the contextual representation of the classification token. A task-specific MLP head ψ_t maps this representation to the two motor-performance outputs:

$$\hat{y}^{(t)} = \psi_t(h_{\mathrm{cls}}). \tag{2}$$

The same output-head architecture is used for all transformer backbones in the experiments so that performance differences mainly reflect the representation model rather than the regression head design.

C. LoRA-Based Task Expansion

The D0-trained model serves as the shared backbone for related motor tasks. To avoid cross-task interference from sequential fine-tuning, the backbone is frozen and each new variant trains only a lightweight LoRA adapter and task-specific regression head.

For a backbone weight matrix $W_0 \in \mathbb{R}^{m\times n}$, LoRA represents the task-specific update as

$$W = W_0 + \frac{\alpha}{r} BA, \tag{3}$$

where $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{m\times r}$ are trainable low-rank matrices initialized such that $BA = 0$, leaving the D0-trained backbone unchanged at initialization. This reduces the trainable parameters per adapted matrix from mn to $r(m+n)$.

Because the backbone remains frozen, each motor variant independently stores its LoRA adapter and regression head, while new tasks leave previously learned parameters unchanged. This task-isolated design enables multiple motor schemas to share one backbone and forms the basis for the schema-guided registry described next.

D. Schema-Guided Registry and Inference

Because each motor variant uses a task-specific LoRA adapter and regression head, deployment requires automatic schema identification and reconstruction of the canonical input format. We therefore introduce the lightweight schema-guided registry shown in Fig. 3.

Each registry entry task names, stores canonical field list, target output names, a reference to the shared backbone, the task-specific LoRA adapter, and the regression head. Given an input query, whose fields may appear in any order, the registry performs the following three-stage procedure:

1) **Schema parsing:** Extract field-value pairs from the input text, separating parameter names from their numerical values.
2) **Canonical reconstruction:** Reorder the extracted field-value pairs according to the training-time field sequence. This step restores the canonical input format expected by the BERT regressor and avoids the distribution shift introduced by shuffled parameter clauses.
3) **Task resolution and adapter activation:** Match the reconstructed field set against the registered task schemas, identify the corresponding motor variant, and activate its LoRA adapter and regression head on the shared backbone.

The schema-guided registry serves two complementary functions. First, it enables deterministic task resolution from the input field set: parameters unique to D1, such as dummy-slot width and height, or to D2, such as magnet-stopper width and height, identify the corresponding schema without manual task selection. Second, it restores the canonical field order used during training before regression. As shown in Section IV-E, directly using shuffled inputs substantially degrades accuracy, whereas canonical reconstruction restores performance to the original input-order baseline. The registry therefore provides both automatic schema routing and consistent input formatting for multi-schema inference.

IV. Experiments

A. Experimental Setup

We evaluate the proposed framework on the three SPM datasets introduced in Section II. D0 is the original design task, while D1 and D2 correspond to the dummy-slot and magnet-stopper variants, respectively. The three datasets contain 9,246, 9,248, and 9,232 samples for D0, D1, and D2. Each dataset is randomly split into training, validation, and test subsets with a 7:1:2 ratio. Unless otherwise stated, all results are reported on the held-out test set.

All models predict two targets, cog_{pp} and emf_{eff}, and are trained with AdamW. We use the weighted RMSE loss, consistent with our previous work [19]:

$$L = \sqrt{\mathrm{mean}\left(|y|^{0.2}(\hat{y} - y)^2\right)}. \tag{4}$$

Transformer-based models are trained for up to 200 epochs with an initial learning rate of 1×10^{-4}, and the best checkpoint is selected according to validation performance.

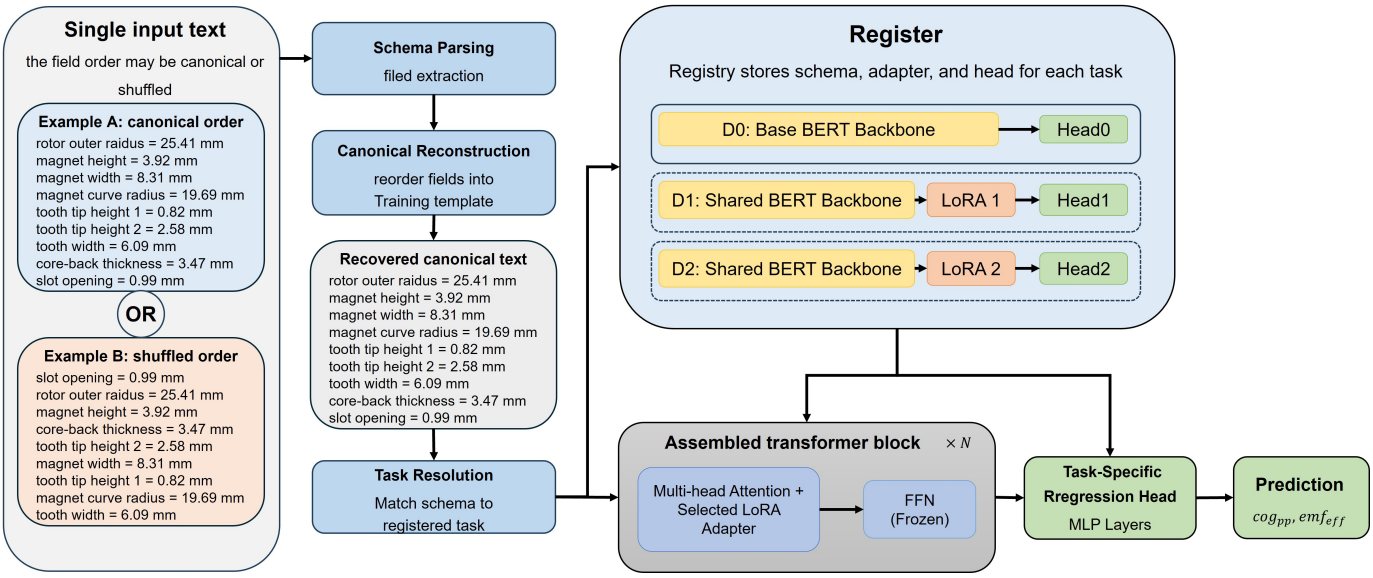

Fig. 3. Schema-guided inference pipeline. The input text may follow either the canonical or a shuffled field order and is first parsed to extract field-value pairs. The extracted fields are then reordered into the canonical training-time sequence. The resulting field set is matched against the registered task schemas, and the registry activates the corresponding LoRA adapter and task-specific regression head on the shared BERT backbone for motor-performance prediction.

The ANN baseline takes the structured numerical parameter vector as input. To provide a well-tuned numerical baseline, its architecture and optimization hyperparameters are selected by Optuna with 50 trials. The final ANN configuration uses five hidden layers with sizes $[512, 512, 128, 64, 64]$, learning rate 1.0×10^{-2}, and weight decay 1.8×10^{-7}. The selected ANN model is then trained for 200 epochs under the same train/validation/test split.

For the text-based surrogate, DistilBERT is used as the encoder backbone. For a fair comparison, all transformer-based models share the same regression head, consisting of three progressively reduced hidden layers with GELU activations and 0.1 dropout, followed by a two-dimensional output for cog_{pp} and emf_{eff}.

In the DistilBERT-LoRA setting, we use LoRA rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. LoRA modules are applied to the query, key, value, and output projections in each self-attention layer. During adaptation to D1 and D2, the shared DistilBERT backbone is frozen, and only the LoRA parameters and the corresponding task-specific regression head are updated.

B. Backbone Comparison on D0

We first compare representative model families on the original D0 task: an Optuna-tuned ANN using structured numerical parameters, a fully fine-tuned Qwen3.5-2B multimodal model with image–text inputs and learnable soft-CLS tokens, a structured parameter-initialized ViT [19], and the proposed DistilBERT-based text model. As shown in Table II, ViT achieves the highest mean R^2 on D0, while DistilBERT obtains the best emf_{eff} performance and matches the best weighted RMSE. The Qwen3.5-2B multimodal baseline, even with soft-CLS tokens for regression-oriented feature aggregation, does not outperform DistilBERT. This suggests that adding image inputs and learnable aggregation tokens to a large multimodal backbone does not necessarily provide additional benefit for this low-dimensional structured regression task. In contrast, directly injecting the motor parameters into ViT classification tokens or serializing them into canonical text descriptions provides a more direct representation of the design variables.

This single-task comparison provides a baseline view of representation quality on the original schema. However, strong D0 performance alone does not guarantee effective adaptation to schema-expanded variants, where new design variables are introduced and the input representation must remain compatible with the D0-trained model. Therefore, we next compare ViT and DistilBERT under the D0-to-D1/D2 adaptation setting.

TABLE II. Backbone comparison on D0.

Method	Weighted RMSE	$R^2_{\mathrm{cog}_{pp}}$	$R^2_{\mathrm{emf}_{eff}}$	Mean R^2
ANN	**0.00061**	0.833	0.981	0.907
Qwen3.5-2B multimodal	0.00072	0.831	0.970	0.899
ViT + param. CLS	0.00062	**0.910**	0.973	**0.941**
DistilBERT text model	**0.00061**	0.859	**0.983**	0.921

C. Cross-Task Adaptation: DistilBERT vs. ViT

We next evaluate cross-task adaptation from D0 to the schema-expanded tasks D1 and D2. DistilBERT-LoRA is compared with ViT-LoRA under the same LoRA configuration, where each model is adapted from its corresponding D0 base model. As shown in Table III, ViT remains competitive on the original D0 task, but DistilBERT-LoRA achieves better performance on both schema-expanded variants.

The gains are consistent on D1 and D2. DistilBERT-LoRA improves the mean R^2 from 0.832 to 0.895 on D1 and from 0.893 to 0.951 on D2. The improvement is mainly driven by cog_{pp}, whose R^2 increases from 0.696 to 0.821

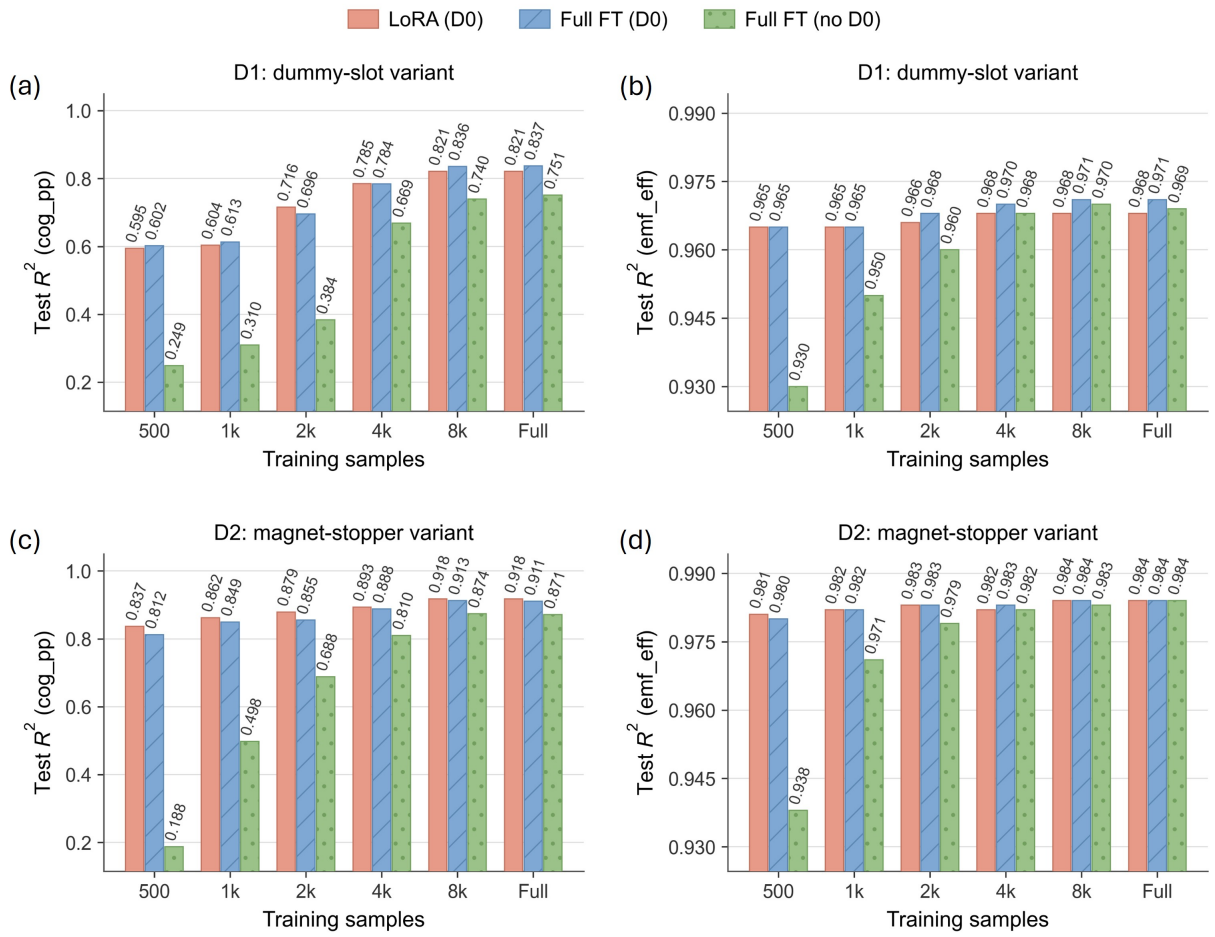

Fig. 4. DistilBERT adaptation under different target-task training sizes on D1 and D2. D0 initialization is most beneficial in the low-data regime, and LoRA remains close to full fine-tuning throughout.

on D1 and from 0.801 to 0.918 on D2. This suggests that image-based encoding may be less effective for the small local schema changes introduced by dummy-slot and magnet-stopper features, because these features modify only limited regions of the motor geometry. In contrast, the text representation explicitly encodes the new fields and their numerical values as parameter-value clauses. Averaged over D0–D2, the DistilBERT family reduces weighted RMSE from 0.00066 to 0.00058 and improves mean R^2 from 0.889 to 0.926.

TABLE III. ViT and DistilBERT across D0–D2.

Data	Method	Weighted RMSE	$R^2_{\text{cog}_{pp}}$	$R^2_{\text{emf}_{eff}}$	Mean
D0	ViT	0.00062	**0.910**	0.973	**0.941**
D0	DistilBERT	**0.00061**	0.859	**0.983**	0.921
D1	ViT-LoRA	0.00075	0.696	0.967	0.832
D1	DistilBERT-LoRA	**0.00058**	**0.821**	**0.968**	**0.895**
D2	ViT-LoRA	0.00060	0.801	**0.985**	0.893
D2	DistilBERT-LoRA	**0.00055**	**0.918**	0.984	**0.951**
Avg.	ViT family	0.00066	0.802	0.975	0.889
Avg.	DistilBERT family	**0.00058**	**0.874**	**0.979**	**0.926**

D. Ablation Study

The proposed framework relies on three key design choices: task isolation, D0-based initialization, and parameter-efficient LoRA adaptation. We first examine whether task-specific adapters are beneficial for simultaneous multi-schema deployment, and then evaluate the roles of D0 initialization and LoRA in data-efficient adaptation to the schema-expanded variants.

1) Task Isolation vs. Sequential LoRA Adaptation: To evaluate task isolation, we first fully fine-tune the backbone on D0. Starting from this D0-trained model, we then sequentially perform LoRA fine-tuning on D1 and D2 using a single shared adapter, evaluating the model on all three tasks after each stage.

Table IV shows substantial task-dependent interference under sequential full fine-tuning. After adaptation to D1, $R^2_{\text{cog}_{pp}}$ on D0 decreases from 0.859 to 0.675, while the D2 score decreases from 0.828 to 0.514. After the subsequent D2 stage, D1 performance drops sharply, with $R^2\text{cog}_{pp}$ decreasing from 0.821 to 0.406. Although performance on some tasks may recover or even improve after later updates, no single

TABLE IV. Sequential LoRA adaptation under the D0→D1→D2 schedule. After each stage, the sequentially updated DistilBERT model is evaluated on all three tasks. Parenthesized arrows indicate changes from the previous stage. **Bold** marks the best R^2 achieved for each task across all stages; these task-wise best values are not attained simultaneously by a single sequential checkpoint.

Training stage	Eval on D0			Eval on D1			Eval on D2		
	$R^2_{\mathrm{cog}_{pp}}$	$R^2_{\mathrm{emf}_{eff}}$	Mean R^2	$R^2_{\mathrm{cog}_{pp}}$	$R^2_{\mathrm{emf}_{eff}}$	Mean R^2	$R^2_{\mathrm{cog}_{pp}}$	$R^2_{\mathrm{emf}_{eff}}$	Mean R^2
D0	0.859	0.983	0.921	0.362	0.960	0.661	0.828	0.978	0.903
D0 → D1	0.675 (↓0.18)	0.981 (↓0.00)	0.828 (↓0.09)	**0.821** (↑0.46)	**0.968** (↑0.01)	**0.895** (↑0.24)	0.514 (↓0.31)	0.980 (↑0.00)	0.747 (↓0.16)
D0 → D1 → D2	**0.928** (↑0.25)	**0.984** (↑0.00)	**0.956** (↑0.13)	0.406 (↓0.42)	0.965 (↓0.00)	0.685 (↓0.21)	**0.918** (↑0.40)	**0.984** (↑0.00)	**0.951** (↑0.20)

sequential checkpoint simultaneously retains the task-wise best performance across all three schemas.

This behavior is undesirable for deployment, where the final surrogate is expected to support all registered motor schemas simultaneously. In contrast, the task-isolated DistilBERT-LoRA models in Table III maintain mean R^2 values of 0.921, 0.895, and 0.951 for D0, D1, and D2, respectively. These results support task-isolated adaptation as an effective strategy for avoiding the cross-task interference observed under sequential fine-tuning.

2) Initialization and Adaptation Efficiency: We further evaluate how the D0-trained model is reused for adapting to the schema-expanded tasks D1 and D2. Three training schemes are compared: (1) LoRA adaptation from the D0-trained DistilBERT model, where the backbone is frozen and only the LoRA adapter and task-specific head are trained; (2) full fine-tuning from the same D0-trained model, where all model parameters are updated on the target task; and (3) full fine-tuning without D0 initialization, where the model is trained only with target-task data. To study data efficiency, we vary the number of target-task training samples from 500 to the full training set. The validation and test sets are kept fixed for all training sizes, and only the target-task training subset is subsampled.

As shown in Fig. 4, the two methods that reuse the D0-trained model, LoRA adaptation and full fine-tuning from D0, clearly outperform the without D0 initialization baseline in the low-data regime. The gap is especially large for cog_{pp}, indicating that the D0-trained encoder already captures useful representations of the shared motor geometry. When only a small number of new FEA samples are available, this prior knowledge helps the model adapt to the dummy-slot and magnet-stopper variants without relearning the common design factors from scratch.

Comparing the two D0-based adaptation methods, LoRA closely follows full fine-tuning across different training sizes on both D1 and D2. This shows that the additional effects introduced by the new schema-specific parameters can be captured effectively by low-rank updates, without modifying the entire DistilBERT backbone. Therefore, the role of LoRA is not to replace the D0 base model, but to adapt the D0-trained representation to each new schema with a small number of task-specific parameters.

This distinction is important for continual deployment. Full fine-tuning from D0 can serve as a strong per-task reference, but it produces a different updated backbone for each new variant, increasing storage and deployment overhead as more schemas are added. In contrast, DistilBERT-LoRA keeps the D0-trained backbone shared and frozen, while each schema only stores its own lightweight LoRA adapter and regression head. This reduces the per-task parameter footprint and enables new motor variants to be added without overwriting or duplicating the shared backbone.

E. Schema-Based Inference

Finally, we evaluate the schema registry under canonical and shuffled input orders. Canonical inputs follow the training template with fields arranged in the predefined sequence, whereas shuffled inputs contain exactly the same field-value pairs in a randomly permuted order. This experiment evaluates two functions of the registry: task resolution from the input field set and reconstruction of the canonical field order expected by the trained DistilBERT regressor.

TABLE V. Schema-based inference under canonical and shuffled inputs.

Motor	Input/mode	Route	$R^2_{\mathrm{cog}_{pp}}$	$R^2_{\mathrm{emf}_{eff}}$	Mean
D0	canonical/off	1.000	0.859	0.983	0.921
D0	shuffle/off	1.000	0.011	-0.123	-0.056
D0	shuffle/on	1.000	0.859	0.983	0.921
D1	canonical/off	1.000	0.821	0.968	0.895
D1	shuffle/off	1.000	-0.211	0.056	-0.078
D1	shuffle/on	1.000	0.821	0.968	0.895
D2	canonical/off	1.000	0.918	0.984	0.951
D2	shuffle/off	1.000	-0.158	-0.041	-0.099
D2	shuffle/on	1.000	0.918	0.984	0.951

Table V shows that task routing is unaffected by clause order. The routing accuracy remains 1.000 for all shuffled inputs because task identity is determined from the set of input fields rather than their sequence. For example, the dummy-slot parameters uniquely identify D1, while the magnet-stopper parameters identify D2.

In contrast, the DistilBERT regressor is highly sensitive to field order. When shuffled inputs are passed directly to the model without canonical reconstruction, the mean R^2 becomes negative for all three tasks. After the registry restores the training-time field sequence, prediction performance returns to the corresponding canonical-input baseline, with mean R^2 values of 0.921, 0.895, and 0.951 for D0, D1, and D2, respectively. These results show that the registry provides two complementary capabilities: deterministic task routing and reconstruction of the input format expected by the trained text regressor.

TABLE VI. Computational and Storage Performance

Metric	Proposed (LoRA)	Full fine-tuning
Trainable parameters	**0.76 M (1.16 %)**	**65.4 M (100 %)**
Backbone	0 (frozen)	65.2 M
LoRA adapter	0.59 M	n/a
Regression head	0.17 M	0.17 M
Stored per task	**2.92 MiB**	**249.37 MiB**
Backbone weights	0 (shared)	248.71 MiB
LoRA adapter	2.26 MiB	n/a
Regression head	0.66 MiB	0.66 MiB
Storage for N topologies	$248.71 + 2.92N$ MiB	$249.37N$ MiB
Latency, $B = 1^{\dagger}$	2.57 ms	2.57 ms
Latency, $B = 128^{\dagger}$	227.6 ms	228.3 ms
Throughput, $B = 128^{\dagger}$	562 samples s^{-1}	561 samples s^{-1}

n/a: full fine-tuning introduces no low-rank adapter.
†NVIDIA RTX 4070, FP32, sequence length 256, batch size B; median of 300 runs. Storage figures are on-disk FP32 checkpoint sizes.

F. Computational and Storage Efficiency

This section evaluates the computational and storage performance of the proposed framework in terms of training cost, memory usage, parameter efficiency, and storage requirements. Using D0 as a representative case, full fine-tuning for 200 epochs requires 3626 s training time and a peak training memory of 10,820 MiB, whereas LoRA under the same setting requires 3429 s and 9793 MiB, corresponding to reductions of 5.4% and 9.5%, respectively. As summarized in Table VI, LoRA requires only 0.76 M trainable parameters and 2.92 MiB of additional storage per topology, compared with 65.4 M parameters and 249.37 MiB for full fine-tuning. This yields an approximately 85$\times$ reduction in additional storage for each newly introduced motor topology.

Beyond these reductions in training cost and storage, LoRA maintains nearly identical inference latency and throughput to full fine-tuning, indicating that its efficiency gains introduce no measurable inference overhead.

V. Conclusion

This paper presented a schema-guided BERT-LoRA framework for continual surrogate modeling of related SPM motor designs. By representing motor parameters as canonical text descriptions, the framework accommodates schema-expanded variants without modifying the shared encoder architecture. Task-specific LoRA adapters enable newly introduced motor variants to be incorporated while keeping the D0-trained backbone frozen, thereby avoiding the cross-task interference observed under sequential full fine-tuning.

Experiments show that D0-based initialization provides an effective prior for adapting to schema-expanded tasks, particularly in the low-data regime, while LoRA achieves performance comparable to full fine-tuning with substantially fewer trainable and stored parameters. The schema-guided registry further enables deterministic task resolution and reconstructs the canonical field order when input clauses are shuffled, allowing the appropriate task-specific adapter and regression head to be selected for inference. Future work will extend the framework to a broader range of motor topologies and investigate confidence-aware handling of incomplete or ambiguous input schemas.

References

[1] J. Kudela and R. Matousek, "Recent advances and applications of surrogate models for finite element method computations: A review," *Soft Computing*, vol. 26, pp. 13 709–13 733, 2022.

[2] M. Cheng, X. Zhao, M. Dhimish, W. Qiu, and S. Niu, "A review of data-driven surrogate models for design optimization of electric motors," *IEEE Transactions on Transportation Electrification*, 2024.

[3] M. Tahkola, J. Keränen, D. Sedov, M. F. Far, and J. Kortelainen, "Surrogate modeling of electrical machine torque using artificial neural networks," *IEEE Access*, vol. 8, pp. 220 027–220 045, 2020.

[4] D.-K. Lim and D.-K. Woo, "A new surrogate-assisted robust multi-objective optimization algorithm for an electrical machine design," *Journal of Electrical Engineering & Technology*, vol. 14, pp. 1247–1254, 2019.

[5] J.-M. Ahn, M.-K. Baek, S.-H. Park, and D.-K. Lim, "Optimal design of ipmsm for ev using subdivided kriging multi-objective optimization," *Processes*, vol. 9, no. 9, p. 1490, 2021.

[6] N. Gabdullin, S. Madanzadeh, and A. Vilkin, "Towards end-to-end deep learning performance analysis of electric motors," in *Actuators*, vol. 10, no. 2. MDPI, 2021, p. 28.

[7] V. Parekh, D. Flore, and S. Schoeps, "Deep learning-based prediction of key performance indicators for electrical machines," *Ieee Access*, vol. 9, pp. 21 786–21 797, 2021.

[8] H. Sasaki, T. Hidaka, and H. Igarashi, "Prediction of ipm machine torque characteristics using deep learning based on magnetic field distribution," *IEEE Access*, vol. 10, pp. 60 814–60 822, 2022.

[9] Y. Shimizu, S. Morimoto, M. Sanada, and Y. Inoue, "Automatic design system with generative adversarial network and convolutional neural network for optimization design of interior permanent magnet synchronous motor," *IEEE Transactions on Energy Conversion*, vol. 38, no. 1, pp. 724–734, 2023.

[10] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.

[11] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[12] S. Yang, M. A. Ali, C.-L. Wang, L. Hu, and D. Wang, "MoRAL: MoE augmented LoRA for LLMs' lifelong learning," *arXiv preprint arXiv:2402.11260*, 2024.

[13] D. Li, Y. Ma, N. Wang, Z. Ye, Z. Cheng, Y. Tang, Y. Zhang, L. Duan, J. Zuo, C. Yang *et al.*, "Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts," *arXiv preprint arXiv:2404.15159*, 2024.

[14] X. Wang, T. Chen, Q. Ge, H. Xia, R. Bao, R. Zheng, Q. Zhang, T. Gui, and X. Huang, "Orthogonal subspace learning for language model continual learning," in *Findings of EMNLP*, 2023.

[15] C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, and M. Lin, "LoraHub: Efficient cross-task generalization via dynamic LoRA composition," in *Proceedings of COLM*, 2024.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[19] S. Sun, Y. Wang, T. Koike-Akino, T. Yamamoto, Y. Sakamoto, and B. Wang, "Electric motor cogging torque prediction with vision transformer models," in *2025 IEEE International Electric Machines & Drives Conference (IEMDC)*. IEEE, 2025, pp. 782–788.