A GA-RL Hybrid Framework for E-Paper Waveform Optimization

1st Ziwei Mo
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
mozw8@mail2.sysu.edu.cn

2nd Hankz Hankui Zhuo
School of Artificial Intelligence
Nanjing University
Nanjing, China
hankz@nju.edu.cn

3rd Zhanwen Zhou
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
zhouzhw26@mail2.sysu.edu.cn

***Abstract*—The calibration of driving waveform parameters is essential for achieving accurate color performance in electronic paper (E-paper) manufacturing. However, this process constitutes a high-dimensional black-box optimization problem, severely constrained by the fact that each real-hardware evaluation requires several minutes. This stringent evaluation budget renders manual tuning inefficient and limits the direct application of intelligent optimization algorithms. While Genetic Algorithms (GA) and Reinforcement Learning (RL) are principled approaches for such black-box optimization, GA can suffer from premature convergence and RL from sparse feedback under extremely limited evaluations. To address these challenges, we propose a novel GA-RL hybrid optimization framework. Our method embeds an RL-based local fine-tuning process, guided by a predictive LAB simulation model, within the global evolutionary cycle of GA. This cooperative integration enables efficient use of the limited evaluation budget. Experimental results demonstrate that the proposed framework significantly outperforms standalone GA or RL, achieving superior optimization efficiency and success rates in practical E-paper production.**

***Index Terms*—genetic algorithm, reinforcement learning, deep learning, electronic paper**

I. Introduction

Recent advances in artificial intelligence have facilitated the widespread integration of machine learning across various industries, promoting the transition of traditional production processes toward intelligent automation. Among various machine learning paradigms, reinforcement learning (RL) has garnered significant attention due to its unique trial-and-error interaction mechanism [1]. This approach enables agents to gradually accumulate experience and optimize strategies through a closed-loop process of "action-feedback-improvement," achieving remarkable success in domains such as game-playing agents, autonomous driving, and robotic control [2].

Electronic paper (E-paper), as a low-power display technology, has been widely adopted in applications such as electronic shelf labels, e-readers, and wearable devices, owing to its paper-like visual experience [3] and extremely low energy consumption [4]. Its color rendering principle, as illustrated in Figure 1, relies on driving waveform parameters [5], which include timing parameters (TP, SR, RP) that control the temporal structure of the waveform, and voltage-level parameters (LUT) that define the electric signal polarity, as shown in Figure 2. Each set of these parameters is parsed into three distinct waveform data segments corresponding to the black, white, and red color channels. These parameters are programmed into the driver chip to generate electric signals applied through upper and lower electrodes, precisely manipulating the movement of charged black, white, and red particles within microcapsules to produce the desired colors [6]. In the manufacturing process, achieving the target color performance requires careful adjustment of these parameters. The objective is to ensure the displayed color matches the standard LAB color values—an internationally recognized color space where L represents perceptual lightness, and a and b denote the color-opponent dimensions, enabling precise quantification of human perceptual differences.

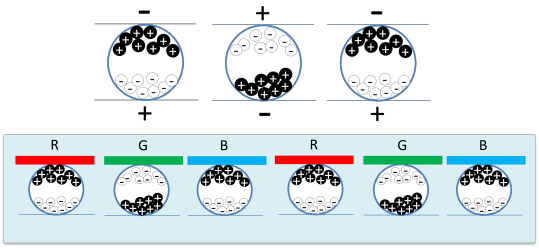

Fig. 1. Electronic paper working principle diagram.

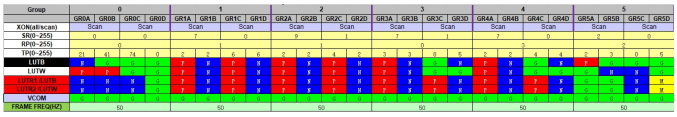

Fig. 2. Waveform parameters diagram.

Currently, the parameter adjustment process remains highly dependent on engineers' expertise. Training a proficient engineer typically requires several months, and calibrating parameters for each product batch can take nearly a full workday. This not only incurs high labor costs but also severely constrains production efficiency. Due to the high dimensionality, strong coupling, and nonlinearity among parameters, as well as the absence of an explicit mathematical model, the task can be essentially formulated as a high-dimensional, nonlinear black-box optimization problem.

To address this challenge, this study investigates the application of two intelligent optimization methods—Genetic Algorithm (GA) [7] and Reinforcement Learning (RL). GA mimics mechanisms of biological evolution through selection, crossover, and mutation to effectively explore the parameter space [8]. RL, in contrast, employs an agent that learns optimal adjustment strategies through iterative interaction with the E-paper system, using color difference as a reward signal [9]. Both methods are well-suited for engineering optimization problems where accurate mathematical models are difficult to establish.

However, a major practical constraint is the time required for real-world evaluations: programming each parameter set into the hardware and measuring the resulting LAB values takes several minutes. This limits the total number of evaluations to several tens or hundreds within a production cycle, posing significant challenges for both GA and RL. GA may suffer from premature convergence or insufficient exploration under limited population or generation sizes. RL, on the other hand, struggles with sparse feedback and inadequate policy learning when environmental interactions are severely restricted.

To overcome these limitations, this paper proposes a GA–RL hybrid optimization framework. The core idea is to embed an RL-based local refinement process within the multi-generational evolutionary cycle of GA. Each generation consists of three stages: First, standard GA operations (selection, crossover, mutation) are performed to generate a new population. Then, each individual in the population undergoes local optimization via RL within a simulation environment constructed by a LAB prediction model. Finally, elite selection is conducted between the original individuals and their RL-optimized counterparts based on real hardware measurements, with the winners serving as seed individuals for the next generation of genetic evolution.

This approach innovatively combines the global exploration capability of GA with the local optimization efficiency of RL, leveraging the low-cost simulation environment to overcome the constraint of limited real evaluations. Through iterative generations, the population quality is progressively improved, accelerating convergence toward the global optimum. The main contribution of this work lies in the novel integration of GA and RL in a cooperative manner to address the problem of E-paper driving waveform optimization—a challenge not previously explored in the literature. This framework provides an effective technical pathway for automated and intelligent parameter tuning under strict evaluation budgets.

II. Related Work

Optimization Methods. Intelligent optimization algorithms [10] and deep reinforcement learning [11] have been widely applied in complex system optimization, offering effective tools for black-box problems. For instance, the Genetic Algorithm (GA) mimics natural evolution by employing selection, crossover, and mutation to explore the global search space [8]. It is particularly suitable for gradient-free optimization and has achieved significant results in engineering tasks such as parameter tuning and scheduling. Deep reinforcement learning algorithms, such as Deep Q-Network (DQN) [12], [13], integrate deep learning with reinforcement learning, enabling agents to learn optimal policies through environmental interactions [9]. These methods have shown remarkable performance in sequential decision-making tasks including game playing and robotic control. Since they do not rely on precise mathematical models, they are well-suited for high-dimensional nonlinear problems like e-paper driving waveform optimization. However, existing studies have primarily focused on simulated environments or scenarios with ample evaluation budgets, leaving the challenges under extremely limited evaluations largely unexplored [14].

Temporal Modeling. Time series analysis models, such as LSTM, Transformer, and TimesNet, provide powerful capabilities for sequence modeling and play a significant role in industrial parameter prediction. Long Short-Term Memory (LSTM) networks capture long-range dependencies through gating structures and are widely used in tasks such as time-series forecasting and anomaly detection [15]. The Transformer model, based on self-attention mechanisms, enables parallel processing of sequential data and has achieved breakthroughs in natural language processing and temporal analysis [16]. TimesNet, as a recently proposed temporal modeling framework, transforms time series into 2D tensors based on periods and captures multi-periodicity and complex temporal patterns through 2D convolutional neural networks, demonstrating strong performance in various time series analysis tasks [17]. These models can be employed to build predictive surrogates for e-paper color display—for example, by mapping driving waveform parameters to LAB color values—thus replacing time-consuming physical measurements.

E-paper Parameter Tuning. Current tuning of e-paper driving waveforms heavily relies on manual expertise, and research on automation remains at an early stage. In e-paper production, parameters such as timing parameters (TP, SR, RP) and voltage-level parameters (LUT) must be adjusted to approximate standard LAB color values [18]. However, due to strong parameter coupling and the absence of an explicit model, the tuning process is time-consuming and susceptible to subjective influence. In industrial practice, manual trial-and-error by engineers remains the dominant approach, highlighting an urgent need for low-cost and efficient automated solutions [19]. Hybrid frameworks that integrate intelligent optimization with simulation surrogates have thus emerged as a promising direction.

III. Methodology

A. Framework Overview

Objective: This paper aims to address the core challenges in e-paper driving waveform parameter optimization—namely, the heavy reliance on manual expertise and the extremely limited budget for physical evaluations. Our goal is to provide a reliable solution for achieving efficient and automated parameter tuning. Specifically, this research focuses on a black, white,

and red (BWR) tricolor electronic paper. Consequently, for any given set of driving waveform parameters, programming them into the hardware and performing measurement yields three LAB color values, corresponding to the display effects of the black, white, and red colors, respectively. Since the production objectives primarily concern luminance and red-green hue, with no specific constraints on the B (blue-yellow) dimension, the optimization target effectively has a 3 (colors) × 2 (L and a dimensions) structure. The ultimate objective of parameter optimization is to identify an optimal parameter set such that the actual measured LAB values of the e-paper closely approximate the target values in the L and a dimensions after programming and measurement.

Overview: To this end, we propose a novel Genetic Algorithm and Reinforcement Learning hybrid optimization (GA-RL) framework. Our research first successfully applied the Genetic Algorithm (GA) and Reinforcement Learning (RL) individually to this problem, confirming the feasibility of both methods for such a complex black-box optimization task. However, the performance improvement of either standalone algorithm reached a bottleneck under strict evaluation constraints. To overcome this limitation, we introduced a neural network-based LAB color prediction model to construct a low-cost, high-efficiency simulation environment. Building on this, the proposed GA-RL hybrid framework effectively combines the advantages of both algorithms: GA is used for global exploration to identify promising parameter regions, and then RL performs fine-tuning of elite solutions within the simulation environment. This framework concentrates the limited real evaluation resources on the most promising candidate parameters. Through a closed-loop process of simulation-real interaction and validation, the strategy and the prediction model are continuously optimized, ultimately achieving significant improvements in color rendering performance under a stringent evaluation budget. Next, we elaborate on the main components.

B. Genetic Algorithm

To address the optimization of driving waveform parameters for electronic paper, this paper employs a Genetic Algorithm (GA) as one of the core optimization methods. As a global optimization algorithm inspired by natural selection and genetic mechanisms, GA is particularly suitable for complex, high-dimensional, nonlinear problems lacking explicit mathematical models [20]. As discussed in Section I, the driving waveform parameters for electronic paper —— including timing parameters (TP, SR, RP) and voltage-level parameters (LUT) – exhibit strong coupling and reside in a high-dimensional space [8], [21]. By maintaining a population of parameter sets (individuals) and simulating an evolutionary process of "selection-crossover-mutation," GA facilitates an effective global search even under a limited evaluation budget. The detailed design and implementation of the GA in this study are described below.

Encoding and Population Initialization: A complete set of driving waveform parameters is encoded as an individual. The initial population is generated directly from a baseline waveform parameter set provided by the e-paper manufacturer for the specific display model. This approach reflects the practical industrial process where engineers typically optimize starting from an existing baseline, and it enhances both practical relevance and convergence efficiency.

Fitness Function: This research focuses on optimizing the driving waveforms for a black, white, and red (BWR) tricolor electronic paper. Consequently, the display performance of all three colors must be considered simultaneously. For each parameter set, three LAB color values for the black, white, and red states are obtained through physical hardware measurement. Since the production specifications impose no strict constraints on the B (blue-yellow) dimension, the optimization primarily targets the L (lightness) and a (red-green) dimensions. Thus, the objective function has a 3 (colors) ×2 (dimensions) structure. The fitness function is defined as the negative of the aggregate Euclidean distance between the measured and target LAB values, considering only the L and A dimensions for each color:

$$\text{fitness} = -\sum_{c\in\{\text{black,white,red}\}} \sqrt{(L_c - L_{c,\text{target}})^2 + (A_c - A_{c,\text{target}})^2} \tag{1}$$

A higher fitness value indicates that the overall color performance of the parameter set is closer to the target specifications.

Selection Operation: A roulette wheel selection strategy is adopted to choose individuals for the crossover pool based on a probability distribution derived from their fitness values. Specifically, the softmax function is applied to the fitness values of all individuals in the population to calculate the selection probability for each individual:

$$P(i) = \frac{\exp(\text{fitness}_i)}{\sum_{j=1}^{N} \exp(\text{fitness}_j)} \tag{2}$$

where N is the population size. Individuals with higher fitness have a greater probability of being selected, thereby promoting the inheritance of desirable traits.

Crossover Operation: To generate new individuals in the parameter space, a group-based crossover strategy is employed. Parent individuals, selected from the previous step, are randomly paired. For each pair, every parameter group (Group) is swapped with a predefined probability p_c. It is important to note that parameter groups containing M and N voltage levels in the LUT are defined as *color-pushing groups*. These groups, typically located at the end of the waveform sequence, significantly influence the final LAB color output. This group-based strategy helps explore new combinations while preserving the structure of potentially beneficial parameter groups.

Mutation Operation: Mutation is crucial for maintaining population diversity and preventing premature convergence. Three mutation operations are designed specifically for the color-pushing groups (i.e., groups containing M and N levels), each applied independently with a probability p_m:

1) **Adjust the ratio of red M and N levels:** Randomly modify the RP parameter within the group to alter the proportion of red particles in the driving waveform.
2) **Randomly increase or decrease the group's duration:** Randomly adjust the SR or RP parameter values to change the duration of the specific color-pushing phase.
3) **Randomly increase or decrease a specific TP column:** Randomly select a column within the group's TP parameter sequence and increment or decrement its value.

As the color-pushing groups substantially impact the LAB output, these targeted mutations effectively guide the search process. Through iterative selection, crossover, and mutation operations, the population evolves over generations, progressively improving its average fitness. Under a constrained budget of real-world evaluations, the designed Genetic Algorithm provides an effective pathway for automating the optimization of driving waveform parameters for tricolor electronic paper.

C. Reinforcement Learning

This paper formulates e-paper driving waveform optimization as a Markov Decision Process (MDP) [22] and employs Deep Q-Network (DQN) [12], [13] for the solution. The key to applying DQN to this problem lies in the customized definition of the core MDP components, including state representation, action space, reward function, and environment. This section elaborates on the design of these components, while the standard algorithmic procedures of DQN (e.g., Q-network update, target network) are well-established and thus not reiterated here.

State Representation: The state vector s_t is constructed by concatenating temporal parameters (flattened to $group\_num\times 7$ dimensions) with measured LAB values (L and a channels for black, white, and red), resulting in a final dimension of $group\_num\times 7+3\times 2$. Voltage-level parameters (LUT) remain fixed during optimization.

Action Space: Discrete actions are employed to enable fine-grained parameter adjustments. For each of the seven temporal parameters per group, three operations are defined: multiply by 2.0 (amplify), multiply by 0.5 (reduce), or add 3 (increment), yielding a total of $group\_num \times 7 \times 3$ possible actions.

Reward Function: The reward function guides the agent to optimize color fidelity. The agent receives a bonus reward R_{bonus} only when the measured LAB values for all three calibration colors (black, white, and red) are within their preset acceptable ranges (i.e., satisfy the LAB constraints). These constraints require the L and a values for each color to be qualified.

If the LAB constraints are not fully satisfied, the reward is a penalty based on the total color difference:

$$r_t = -\sum_{c\in\{\text{B,W,R}\}} \sqrt{(L_c - L_{c,\text{target}})^2 + (a_c - a_{c,\text{target}})^2} \quad (3)$$

Otherwise, the agent receives a positive constant bonus:

$$r_t = R_{\text{bonus}} \quad (4)$$

Note the R_{bonus} is a tunable hyperparameter; in our experiments, it is typically set to 10. Furthermore, the LAB constraints used during training are intentionally set to be more stringent than those required in the production environment to encourage a more robust and precise calibration.

Environment and Implementation: The agent interacts with a simulation environment based on the LAB prediction model (or physical hardware). Training is initialized with manufacturer-provided baseline parameters, and exploration is conducted via an ε-greedy policy. The DQN architecture utilizes fully-connected layers to map states to action values, with experience replay employed to stabilize training. This approach facilitates efficient policy learning under limited evaluation budgets by focusing on parameter adjustments that enhance color performance.

D. LAB Simulation Function

To overcome the bottleneck of high-cost physical evaluations, this paper constructs a neural network-based LAB color prediction model that serves as an efficient simulation environment. The core objective of this model is to learn the complex nonlinear mapping from waveform parameter encodings to displayed colors, enabling low-cost and rapid evaluation of color performance for different parameter combinations within the simulation environment. The computational flow of the LAB simulation function adopts an innovative differential prediction architecture, with its mathematical data flow described below.

The model takes three inputs: a reference waveform parameter encoding C_a (for which the actual LAB values are known from physical measurement), the corresponding actual measured LAB values $\mathbf{LAB}_a$, and a target waveform parameter encoding C_b for which the LAB values are to be predicted.

Each waveform parameter encoding (represented as tabular encodings shown in figure 2) is parsed into three waveform data sequences (temporal signals) corresponding to black (B), white (W), and red (R) colors respectively:

$$X_a = [X_a^B, X_a^W, X_a^R], \quad X_b = [X_b^B, X_b^W, X_b^R]$$

where each waveform data is a multidimensional time series. Since the optimization objective primarily focuses on luminance (L) and red-green chroma (a) dimensions, the actual measured input from C_a is represented as the matrix:

$$\mathbf{LAB}_a = \begin{bmatrix} L_a^B & a_a^B \\ L_a^W & a_a^W \\ L_a^R & a_a^R \end{bmatrix}$$

The model first processes the three-color waveform data parsed from parameter encodings C_a and C_b through a weight-sharing encoder to obtain their hidden representations. The encoding process can be expressed as:

$$\mathbf{h}_a = \text{Encoder}(X_a^B, X_a^W, X_a^R), \quad \mathbf{h}_b = \text{Encoder}(X_b^B, X_b^W, X_b^R)$$

The encoder can employ various temporal modeling architectures such as Multi-Layer Perceptron (MLP) [23], Long Short-Term Memory (LSTM) [15], Transformer [16], or TimesNet

[17] to effectively capture temporal dependencies and complex patterns in the waveform data. The predictive capabilities of different encoder architectures will be compared in the subsequent experimental section.

The model then computes the difference between the two hidden representations:

$$\Delta \mathbf{h} = \mathbf{h}_b - \mathbf{h}_a$$

This difference vector $\Delta \mathbf{h}$ encodes the variation information from parameter encoding C_a to C_b. Subsequently, $\Delta \mathbf{h}$ is concatenated with the actual measured LAB matrix of parameter encoding C_a and fed into a Multi-Layer Perceptron (MLP) for regression prediction:

$$\Delta \mathbf{LAB}_{\text{pred}} = \text{MLP}\left([\Delta \mathbf{h},\ \mathbf{LAB}_a]\right)$$

where the MLP output is the predicted LAB value variation:

$$\Delta \mathbf{LAB}_{\text{pred}} = \begin{bmatrix} \Delta L^B & \Delta a^B \\ \Delta L^W & \Delta a^W \\ \Delta L^R & \Delta a^R \end{bmatrix}_{\text{pred}}$$

Finally, the predicted LAB matrix for waveform parameter encoding C_b is obtained by adding the predicted variation to the actual measured values of parameter encoding C_a:

$$\mathbf{LAB}_{b,\text{pred}} = \mathbf{LAB}_a + \Delta \mathbf{LAB}_{\text{pred}} = \begin{bmatrix} L_a^B + \Delta L^B & a_a^B + \Delta a^B \\ L_a^W + \Delta L^W & a_a^W + \Delta a^W \\ L_a^R + \Delta L^R & a_a^R + \Delta a^R \end{bmatrix}_{\text{pred}}$$

This differential prediction architecture offers significant advantages. It essentially learns relative changes in the color space rather than absolute values, enabling the model to focus on color effects induced by parameter encoding variations, which potentially improves generalization and prediction accuracy for fine-grained parameter adjustments. Furthermore, this architecture naturally incorporates known baseline information (actual measurements of parameter encoding C_a), anchoring predictions around a reliable reference point.

The model is trained to minimize the error between the predicted value $\mathbf{LAB}_{b,\text{pred}}$ and the actual measured LAB matrix $\mathbf{LAB}_b$ corresponding to waveform parameter encoding C_b, using an appropriate loss function such as Mean Squared Error (MSE):

$$\mathcal{L} = \|\mathbf{LAB}_{b,\text{pred}} - \mathbf{LAB}_b\|^2$$

The model is trained on a dataset containing historical waveform parameter encodings and their corresponding three-color LAB measurements, enabling it to accurately simulate the physical color rendering process of the electronic paper.

Within the proposed GA-RL hybrid framework, the trained LAB simulation function primarily serves to construct a high-fidelity simulation environment for training the Reinforcement Learning (RL) agent. This environment allows the RL agent to conduct extensive, low-cost trial-and-error learning—exploring parameter adjustment strategies and iteratively refining its policy—without requiring expensive physical measurements at each step. The elite parameter candidates identified by the Genetic Algorithm (GA) through global exploration serve as high-quality starting points for the RL agent's fine-tuning process within this simulation. Ultimately, only the parameters optimized by the RL agent in the simulation environment are selected for final validation using the limited physical evaluation resources. By offering a reliable and efficient training platform for RL, the LAB simulation function effectively decouples the learning process from physical hardware constraints, thereby significantly expanding the effective "exploration budget" and enhancing overall optimization efficiency under strict evaluation limitations.

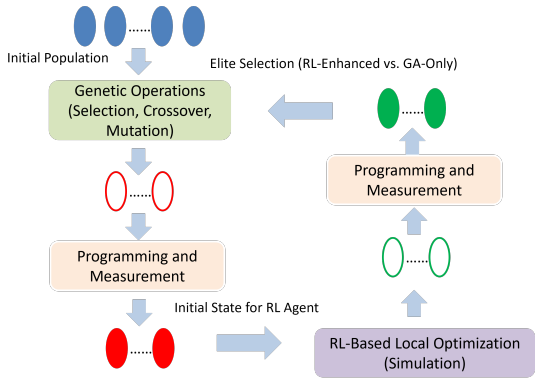

Fig. 3. GA-RL hybrid algorithm workflow.

E. GA-RL Hybrid Framework

The GA-RL hybrid framework integrates Reinforcement Learning (RL) as a local refinement mechanism within the generational cycle of a Genetic Algorithm (GA), aiming to enhance population quality while operating under stringent real hardware evaluation budgets. A key enabler is the LAB simulation function, which provides a low-cost environment for extensive RL training. As illustrated in Figure 3, the process iterates through generations, each comprising a "GA Evolution → RL Enhancement" cycle.

Genetic Algorithm Evolution. Each generation begins with the elite population from the previous cycle. A standard GA iteration is executed: selection based on real-evaluated fitness, crossover to recombine parameters, and mutation—with emphasis on critical "color-pushing parameter groups"—to maintain diversity. This produces a new candidate population, consistent with the method outlined in Section III-B.

Reinforcement Learning Local Optimization. This is the core innovation. Before progressing to the next generation, every individual in the new GA population undergoes local optimization via RL within the LAB simulation environment (Section III-D). Each individual serves as the initial state for a DQN-based agent (Section III-C), which performs a sequence of parameter adjustments. After each action, the simulation predicts the resulting LAB values and computes a reward. The agent's objective is to find a trajectory of adjustments that improves color performance locally, all without consuming real evaluation resources. This process yields a refined counterpart for each original GA individual.

Algorithm 1 GA-RL Hybrid Optimization Framework

Require:
- Initial parameter set P_0 (manufacturer-provided)
- LAB constraints C_{LAB} (for black, white, and red colors)
- Maximum runtime $T_{\max}$

Ensure:
- Optimized parameters x_{best}
- Flag found(indicating whether constraints are satisfied)

$P \leftarrow P_0$ ▷ Initialize population
Compute fitness for all $x \in P$
$x_{\text{best}} \leftarrow \arg\max_{x \in P} \text{fitness}(x)$
$t_{\text{start}} \leftarrow$ current time()
$found \leftarrow$ False
while current time() $- t_{\text{start}} < T_{\max} \wedge \neg found$ **do**
  Step 1: GA Evolution
  $P_{\text{GA}} \leftarrow \text{GA\_Evolve}(P)$
  Compute fitness for all $x \in P_{\text{GA}}$
  Step 2: RL Local Optimization (Simulation)
  $P_{\text{RL}} \leftarrow \{\}$
  for $x_i \in P_{\text{GA}}$ **do**
    $x_i' \leftarrow \text{RL\_LocalOptimize}(x_i)$
    $P_{\text{RL}}.\text{add}(x_i')$
  end for
  Compute fitness for all $x \in P_{\text{RL}}$
  Step 3: Elite Selection & Termination Check
  $P_{\text{next}} \leftarrow \{\}$
  for $i = 1$ to $|P_{\text{GA}}|$ **do**
    if $\text{fitness}(P_{\text{RL}}[i]) > \text{fitness}(P_{\text{GA}}[i])$ **then**
      $x_{\text{candidate}} \leftarrow P_{\text{RL}}[i]$
    else
      $x_{\text{candidate}} \leftarrow P_{\text{GA}}[i]$
    end if
    $P_{\text{next}}.\text{add}(x_{\text{candidate}})$
    if $\text{LAB}(x_{\text{candidate}})$ satisfies C_{LAB} **then**
      $found \leftarrow$ True
      $x_{\text{best}} \leftarrow x_{\text{candidate}}$
      break
    end if
    if $\text{fitness}(x_{\text{candidate}}) > \text{fitness}(x_{\text{best}})$ **then**
      $x_{\text{best}} \leftarrow x_{\text{candidate}}$
    end if
  end for
  $P \leftarrow P_{\text{next}}$
end while
return $(x_{\text{best}}, found)$

Elite Selection and Population Update. Following the RL enhancement, both the original GA-generated individuals and their RL-optimized counterparts are programmed and measured on real hardware to obtain their actual LAB values and fitness scores. A pairwise elite selection is then performed: for each corresponding pair, the individual demonstrating superior real-world performance is retained. This selected elite becomes a seed individual for the next generation of the genetic algorithm. Crucially, during this evaluation phase, if any candidate solution is found to satisfy all predefined LAB constraints, the optimization loop terminates immediately, returning this feasible solution. This early termination mechanism enhances optimization efficiency upon finding a satisfactory result. Otherwise, the algorithm proceeds by forming the next generation entirely from the selected elites. This update strategy ensures that the population evolves based on individuals that have been both locally refined through simulation and rigorously validated in reality, effectively concentrating real evaluations on the most promising regions of the parameter space.

TABLE I
TARGET LAB VALUE CONSTRAINTS

Color	L Constraint	A Constraint
Black	$L < 14$	$A < 4$
White	$L > 65$	$A < 0$
Red	$L > 27$	$41 < A < 45$

IV. EXPERIMENT

In this section, we conduct a comprehensive evaluation of the proposed methods. First, we present the performance comparison of the standalone Reinforcement Learning (RL) algorithm, the Genetic Algorithm (GA), and the GA-RL hybrid framework under actual production constraints. We then provide a detailed comparative analysis of different neural network architectures for the LAB simulation function, which serves as a core component of the hybrid framework's simulation environment.

A. Experiment Settings

The experiments were conducted using the P266010 E-paper model within a temperature range of 25°C to 30°C. The target LAB value constraints are specified in Table I. A parameter set is considered valid—indicating successful tuning—only if the measured L and a values for all three colors (black, white, and red) simultaneously satisfy these constraints. For all parameter tuning algorithms (RL, GA, and the hybrid framework), the optimization process was initialized using the same baseline waveform parameter set, which comprises four distinct waveform parameter groups.

For the LAB simulation function, the training dataset was constructed from historical tuning records of 20 previous tuning tasks performed on the P266010 model. Each task contained approximately 100 distinct waveform parameter sets and their corresponding measured LAB values. Following the differential prediction architecture described in Section III-D, where the model takes waveform parameter encodings C_a and C_b along with the measured LAB values of C_a to predict the LAB values of C_b, the dataset construction process specifically leveraged this input-output relationship. For each tuning task, every possible pair of distinct parameter sets (C_a, C_b) was used to create a training sample, resulting in approximately $\binom{100}{2} = 19,900$ samples per task. With 20 historical tuning

TABLE II
SUCCESS RATE COMPARISON OF DIFFERENT ALGORITHMS (%)

Algorithm	30 minutes	4 hours	12 hours	24 hours
GA	10	35	60	95
RL-Standard	15	20	30	40
RL-Simple	10	20	35	60
GA-RL Hybrid	**45**	**75**	**90**	**100**

tasks, this methodology yielded a comprehensive dataset of approximately 199,000 samples, ensuring robust training of the LAB simulation function. The model was trained following the methodology detailed in Section III-D. In the evaluations conducted in the actual production environment, the Transformer architecture was selected as the encoder for the LAB simulation function based on its superior performance in our comparative analysis.

B. Performance Evaluation in Production Environment

To comprehensively assess the effectiveness of the proposed methods under practical constraints, we conducted extensive experiments in the actual production environment. The primary evaluation metric was the *success rate* of completing the parameter tuning task—defined as finding a valid parameter set that satisfies all LAB constraints specified in Table I—within different time intervals: 30 minutes, 4 hours, 12 hours, and 24 hours. These timeframes reflect realistic production cycle requirements and the practical time constraints faced by engineers during manual tuning processes.

Given the severe limitation on the number of real hardware evaluations in production settings, we identified that the standard Reinforcement Learning approach described in Section III-C, with its extensive action space of $group\_num \times 7 \times 3$ actions, might face challenges in effective policy learning within the limited interaction budget. To address this, we developed a simplified RL variant (denoted as RL-Simple) that focuses optimization efforts exclusively on the temporal parameters of the final waveform group—a critical segment identified through empirical knowledge as having substantial influence on the final color performance. This reduction yields a manageable action space of $7 \times 3 = 21$ discrete actions, significantly improving learning efficiency under constrained evaluations.

We compared four distinct approaches: the standard Genetic Algorithm (GA), the full Reinforcement Learning (RL-Standard) method, the simplified RL approach (RL-Simple), and our proposed GA-RL hybrid framework. Each algorithm was evaluated through 20 independent tuning trials to ensure statistical significance, with all experiments conducted under identical initial conditions and evaluation budgets.

The experimental results presented in Table II reveal significant performance variations among the four evaluated algorithms. Our proposed GA-RL hybrid framework consistently achieved the highest success rates across all time intervals, reaching 45% at 30 minutes, 75% at 4 hours, 85% at 12 hours, and achieving perfect 100% success within 24 hours. Particularly noteworthy is the framework's performance in the 30-minute interval, where it attained 45% success despite being limited to only 1-2 complete "GA evolution-RL optimization" cycles, given that each parameter programming and measurement operation requires approximately 2 minutes.

The Genetic Algorithm (GA) demonstrated strong long-term performance, reaching 95% success after 24 hours, significantly outperforming both RL variants. This robust performance confirms GA's effectiveness in global exploration of the high-dimensional parameter space. It should be noted that the differences in 30-minute success rates between GA and the RL variants (10% versus 15% and 10%) are not statistically significant, indicating that all algorithms face similar exploration challenges under severe time constraints.

Both RL variants showed comparatively lower performance than the population-based optimization methods. The standard RL method achieved a maximum success rate of 40% after 24 hours, while the simplified RL-Simple variant reached 60%. This performance gap suggests that despite improvements through action space reduction in RL-Simple, RL algorithms still face challenges in learning efficiency under limited interaction budgets.

The superior performance of the GA-RL hybrid framework stems from its unique synergistic mechanism: GA handles global exploration while RL performs local fine-tuning in the simulation environment. The framework's ability to achieve 45% success within 30 minutes - corresponding to only approximately 15 hardware evaluations - demonstrates its effectiveness in rapidly identifying promising parameter regions.

These results conclusively show that under strict time constraints and limited hardware evaluation budgets, the hybrid framework combining evolutionary algorithms with reinforcement learning, supported by an accurate simulation environment, effectively addresses the high-dimensional black-box optimization problem of E-paper waveform parameters. The GA-RL hybrid framework not only achieves higher ultimate success rates but, more importantly, demonstrates significant advantages in time-critical production environments.

C. Evaluation of LAB Simulation Function

The accuracy of the LAB simulation function is critical to the effectiveness of the proposed GA–RL hybrid framework, as it forms the core of the simulation environment in which reinforcement learning performs local optimization. To systematically evaluate the predictive performance of different encoder architectures, we conducted a comparative study employing four distinct neural network models: Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), TimesNet, and Transformer.

The dataset described in Section IV-A, comprising approximately 199,000 samples, was used for model training and evaluation. The dataset was partitioned in an 80-10-10 ratio for training, validation, and testing sets, respectively, with data from the same tuning task kept within the same split to prevent information leakage. Model performance was assessed using Mean Squared Error (MSE) on the test set.

TABLE III
PREDICTION PERFORMANCE COMPARISON (MSE)

Encoder	Black		White		Red	
	L	a	L	a	L	a
MLP	2.34	1.87	2.12	1.92	2.25	2.45
LSTM	0.99	0.85	0.95	0.78	1.62	1.97
TimesNet	0.23	0.41	0.21	0.30	0.95	1.28
Transformer	**0.08**	**0.22**	**0.08**	**0.12**	**0.43**	**0.65**

As shown in Table III, the Transformer-based encoder achieved the lowest MSE values across all color channels and both output dimensions, demonstrating a clear advantage, especially for the more challenging red color parameters. This superior performance is likely due to the self-attention mechanism's ability to capture long-range dependencies and complex temporal patterns in the waveform data.

TimesNet ranked second overall, showing competitive results with MSE values slightly higher than those of the Transformer. LSTM exhibited moderate performance, surpassing the MLP but falling short of the more advanced temporal architectures. The MLP encoder, while computationally the simplest, yielded the highest prediction errors across the board.

A consistent trend observed across all models is that prediction accuracy is higher for black and white colors than for red, suggesting that red color parameters are inherently more difficult to predict in this E-paper waveform parameters tuning context. The stability of the performance ranking across different color channels underscores the generalizability of these findings.

These results confirm the rationale behind selecting Transformer as the encoder architecture for the LAB simulation function in our production environment evaluations. The superior predictive accuracy of the Transformer-based model ensures high-fidelity simulation, which in turn enhances the effectiveness of the RL-based local optimization within the GA-RL hybrid framework.

V. CONCLUSION

In this paper, we proposed a GA-RL hybrid optimization framework to address the challenge of E-paper driving waveform optimization under extremely limited evaluation budgets. By integrating genetic algorithms for global exploration with reinforcement learning for local refinement through a LAB color prediction model, our framework effectively overcomes the constraints of traditional manual tuning approaches. Experimental results demonstrate that the proposed method significantly outperforms individual optimization methods (including both GA and RL), achieving superior performance in practical production environments. This work provides the first automated solution for E-paper parameter tuning, breaking away from conventional manual approaches, and offers potential applications in other domains facing similar high-dimensional optimization challenges with restricted evaluations.

REFERENCES

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[3] B. Comiskey, J. D. Albert, H. Yoshizawa, and J. Jacobson, "An electrophoretic ink for all-printed reflective electronic displays," *Nature*, vol. 394, no. 6690, pp. 253–255, 1998.

[4] M. Pitt, R. Zehner, K. Amudson, and H. Gates, "53.2: Power consumption of micro-encapsulated display for smart handheld applications," *Sid Symposium Digest of Technical Papers*, vol. 33, 05 2002.

[5] W.-C. Kao, W.-T. Chang, and J.-A. Ye, "Driving waveform design based on response latency analysis of electrophoretic displays," *Journal of Display Technology*, vol. 8, no. 10, pp. 596–601, 2012.

[6] R. W. Zehner, K. Amundson, A. N. Knaian, B. Zion, M. T. Johnson, and G. Zhou, "20.2: Drive waveforms for active matrix electrophoretic displays," *SID Symposium Digest of Technical Papers*, vol. 34, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:111315813

[7] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addison-Wesley Pub. Co.,*, 1989.

[8] H. D. Garis, *Introduction to Evolutionary Computing*. Springer,, 2003.

[9] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1–22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522000288

[10] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, p. 268–308, Sep. 2003. [Online]. Available: https://doi.org/10.1145/937503.937505

[11] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: An overview," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Y. Bi, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2018, pp. 426–440.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *Computer Science*, 2013.

[13] Volodymyr, Mnih, Koray, Kavukcuoglu, David, Silver, Andrei, A, Rusu, and Joel, "Human-level control through deep reinforcement learning." *Nature*, 2015.

[14] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Mach. Learn.*, vol. 110, no. 9, p. 2419–2468, Sep. 2021. [Online]. Available: https://doi.org/10.1007/s10994-021-05961-4

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[17] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *International Conference on Learning Representations*, 2023.

[18] W. Zeng, Z. Yi, X. Zhou, Y. Zhao, H. Feng, J. Yang, L. Liu, F. Chi, C. Zhang, and G. Zhou, "Design of driving waveform for shortening red particles response time in three-color electrophoretic displays," *Micromachines*, vol. 12, p. 578, 05 2021.

[19] J. C. Heikenfeld, P. S. Drzaic, J. Yeo, and T. Koch, "Review paper: A critical review of the present and future prospects for electronic paper," *Journal of the Society for Information Display*, vol. 19, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:18340648

[20] K. Deb, "Evolutionary algorithms in engineering applications," *Springer Science & Business Media*, 2003.

[21] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[22] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.