International Journal of Research in Computing

REVIEW ARTICLE
Volume 5, Issue 2 (2026)

Learning-Rate Scheduling for Neural Network Training: A Scoping Review of ReduceLROnPlateau, Cosine Annealing with Warm Restarts, Cyclical Learning Rates, and Linear Warmup with Linear Decay

SPGE Illukkumbura[1]

[1]Department of Computing and Information Systems, Faculty of Applied Sciences, Wayamba University of Sri Lanka

[1]gihanerangassck99@gmail.com | https://orcid.org/0009-0007-3849-1529

Abstract

This scoping review presents a structured synthesis of learning-rate scheduling for neural network training. The learning rate is a consequential hyperparameter in gradient-based neural network training because it mediates the trade-off between early exploration and late-stage convergence precision. The review focuses on four widely used learning-rate scheduling paradigms: (1) ReduceLROnPlateau (RLROP), a feedback-driven reactive scheduler; (2) Stochastic Gradient Descent with Warm Restarts (SGDR), a periodic cosine-shaped schedule; (3) Cyclical Learning Rates (CLR), an oscillatory triangular or exponential schedule; and (4) Linear Warmup with Linear Decay (LW+LD), a two-phase schedule widely used with transformer models. We use a common notation as a pedagogical organizing device, distinguish formal results from empirical tendencies and heuristic analogies, and summarize how each method relates to classical step-size conditions and adaptive-optimizer stability. The quantitative tables are explicitly treated as heterogeneous literature summaries rather than controlled head-to-head benchmarks; they are retained only to document source-specific evidence and practical signals. The article includes a scoping-search log, a quality-assessment rubric, caveats on benchmark comparability, hyperparameter-sensitivity diagnostics, failure-mode checks, and an AI tool usage and human contribution statement. The resulting scheduler-selection framework is intended as practical guidance, not as a statistically ranked performance claim.

KEYWORDS

Adaptive learning rate, Cosine annealing, Cyclical learning rates, Deep learning optimization, Learning-rate warmup, ReduceLROnPlateau, Stochastic gradient descent.

ARTICLE HISTORY

Published: 16 July 2026

DATA/CODE AVAILABILITY

The datasets are publicly available at literature sources cited throughout the manuscript. The software code is publicly available at `https://github.com/GihanIllukkumbura/lr_scheduling_tutorial`.

SDG ALIGNMENT

SDG 4
SDG 9

1 Introduction

This manuscript is written as a scoping review and synthesis. It consolidates heterogeneous evidence and practitioner signals for learning-rate scheduling and does not report new controlled experiments or a statistically validated head-to-head ranking among schedulers.

A. Background and Motivation

The optimization of deep neural networks remains among the most actively studied problems in machine learning [1], [2]. Central to every gradient-based training algorithm is a single scalar quantity the learning rate η that controls the magnitude of parameter updates in response to the estimated loss gradient. For stochastic gradient descent (SGD), the fundamental parameter update at step t is:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L(\theta_t;\, B_t)$$

where $\theta_t \in \mathbb{R}^d$ is the parameter vector, L is the empirical loss, and B_t is the randomly sampled mini-batch at step t. The consequences of mis specifying η_t are severe: values that are too large cause oscillatory divergence [1], while values that are too small yield impractically slow convergence and susceptibility to entrapment in suboptimal loss-landscape regions saddle points, narrow valleys, and sharp minima [3]. For a strongly convex quadratic objective with Hessian eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$, the theoretically optimal fixed learning rate is:

$$\eta^* = \frac{2}{\lambda_{\max} + \lambda_{\min}}$$

yielding linear convergence with rate $\rho = [(\kappa - 1)/(\kappa + 1)]^2$ where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the Hessian condition number. In practice, deep network loss surfaces are non-convex, high-dimensional, and non-stationary; neither $\lambda_{\max}$ nor κ is available, and both change continuously throughout training. A fixed η is thus always a crude compromise. This fundamental tension motivates learning rate scheduling the systematic variation of η_t over training. The theoretical necessity of decay is established by the Robbins–Monro conditions: convergence of stochastic approximation requires the step-size sequence $\{\eta_t\}$ to satisfy

$$\sum_{t=1}^{\infty} \eta_t = \infty \qquad \text{and} \qquad \sum_{t=1}^{\infty} \eta_t^2 < \infty.$$

These dual constraints divergence (to allow progress from any starting point) and square-summability (to suppress noise-induced oscillations) form the theoretical scaffolding on which all modern schedules are built.

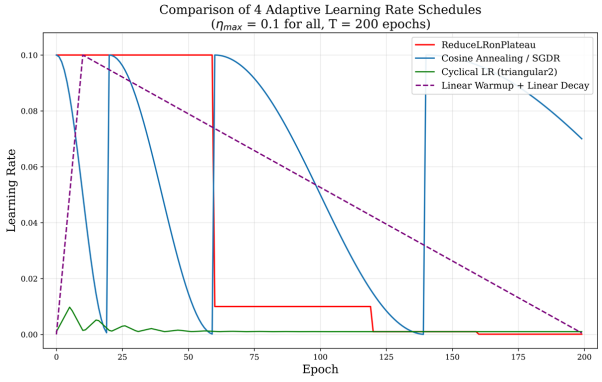

Figure 1: Overview of the four adaptive learning rate schedules analyzed in this paper.

ReduceLROnPlateau (red) produces a staircase determined by training dynamics; SGDR (blue) generates smooth cosine arches with periodic restarts; CLR (green) oscillates as a triangular wave with halving amplitude (triangular2 policy); LW+LD (purple, dashed) rises linearly during warmup then decays monotonically.

B. Problem Statement and Scope

This paper addresses the following review and scoping question: What practical guidance can be drawn from the mathematical forms, documented empirical uses, hyperparameter sensitivities, and known failure modes of ReduceLROnPlateau, Cosine Annealing with Warm Restarts, Cyclical Learning Rates, and Linear Warmup with Linear Decay, while respecting the limits of heterogeneous published evidence. The four strategies represent distinct paradigmatic classes:

1. ReduceLROnPlateau (RLROP) – feedback-based adaptive schedules (reactive control)
2. SGDR / Cosine Annealing – periodic annealing schedules (proactive, cyclic)
3. Cyclical Learning Rates (CLR) – oscillatory schedules (bounded triangular cycles)
4. Linear Warmup + Linear Decay (LW+LD) – warmup-based schedules (two-phase linear)

C. Contributions

This paper presents a scoping review and structured synthesis rather than a new algorithm, new convergence theorem, or controlled benchmark study. We consolidate existing work across four learning-rate scheduling paradigms and make the evidential status of each claim explicit. Specifically, we:

- Use a common notation to organize the four schedulers as a pedagogical scaffold, without claiming that the notation by itself constitutes a new theoretical framework.
- Classify statements about convergence, warmup stability, saddle-point escape, and flat-minima bias as formal results, empirical tendencies, or heuristic interpretations.
- Present four implementation-oriented pseudocode algorithms describing the operational procedure for each method, aligned with common framework conventions in Keras and PyTorch.
- Compile benchmark signals from original publications with explicit source, setting, and comparability caveats rather than treating them as a single controlled leaderboard.
- Provide a heuristic scheduler-selection matrix, hyperparameter-sensitivity diagnostics, and failure-mode checks for practitioners.

Accordingly, the paper narrows its claims: compiled benchmark values are not used to infer a statistically valid ranking among schedulers, and asymptotic convergence statements are not presented without assumptions or citations.

D. Positioning Statement and Non-Claims

Positioning statement. This manuscript is a review article written as a scoping review and synthesis. It does not claim (i) a new learning-rate scheduler algorithm, (ii) a new general convergence theorem, (iii) new controlled benchmark experiments, or (iv) a meta-analysis with pooled effect sizes. All quantitative values are reported as source-specific signals with comparability caveats.

E. Taxonomy of Learning-Rate Scheduling Approaches

To strengthen synthesis value, we organize learning-rate scheduling approaches into higher-level categories (feedback-driven, cyclical, restart-based, warmup-based, and hybrid/adaptive) and summarize typical application contexts and key hyperparameters. A conceptual taxonomy diagram is provided in Appendix (Figure A3).

Table 1: High-level taxonomy of learning-rate scheduling approaches and typical usage contexts.

Category	Representative schedules	Control signal	Typical domains	Key hyperparameters	Notes / caveats
Feedback-driven	ReduceLROn Plateau (RLROP)	Validation metric stagnation	When horizon is unknown; stable validation	factor f, patience p, cooldown d, min_delta, min_lr	Sensitive to metric noise; monitoring choice matters.
Cyclical	CLR triangular/ (triangular2/) exp_range	Predefined periodic LR waveform	CNN training; fast early progress; limited budgets	$\eta_{\min}$, $\eta_{\max}$, step_size (half-cycle), cycle count	Bounds should be set via an LR range test to avoid divergence.
Restart-based	Cosine annealing with warm restarts (SGDR)	Predefined cosine cycles with restarts	Longer training where exploration + snapshot diversity help	T_0, T_{mult}, $\eta_{\max}$, $\eta_{\min}$	Cycle design influences benefit; restarts can destabilize if $\eta_{\max}$ too high.
Warmup-based	Linear warmup + decay (LW+LD); warmup + cosine	Monotone decay with early warmup	Transformer pretraining/fine tuning; stability-critical regimes	warmup_steps or warmup_ratio; peak LR; decay length	Warmup mitigates early instability; peak LR remains dominant risk factor.
Hybrid / adaptive	Warmup+cosine+ restarts; auto tuned schedules	Mixed or data driven control	Foundation/ diffusion models; automatic selection research	model/optimizer dependent	Evidence base is emerging; avoid universal ranking claims.

2 Literature review

A. Classical Decay Schedules

Before the era of adaptive scheduling, practitioners relied on fixed functional forms of learning rate decay. Bottou [10] analyzed the time-based decay $\eta_t = \eta_0/(1 + \lambda t)$ and polynomial decay

$\eta_t = \eta_0 t^{-\alpha}$ ($\alpha \in (0.5, 1]$), establishing that both satisfy the Robbins-Monro conditions. Goodfellow et al. [1] catalogued step decay $\eta_t = \eta_0 \alpha^{\lfloor t/k \rfloor}$ (the ResNet standard [11]), exponential decay $\eta_t = \eta_0 e^{-\lambda t}$, and piecewise-linear decay, noting their widespread empirical success despite limited theoretical justification.

B. Adaptive Per-Parameter Optimizers

Duchi et al. [12] introduced AdaGrad, which adapts per-parameter learning rates via accumulated squared gradients effectively implementing parameter-level decay, but suffering from monotonically shrinking rates. Tieleman and Hinton proposed RMSprop [13] to address this via exponential moving averages. Kingma and Ba [14] unified these into Adam, pairing bias-corrected first and second moment estimates. Adam's global learning rate α remains a critical hyperparameter; adaptive per-parameter scaling does not eliminate the need for a well-designed global schedule. Loshchilov and Hutter [15] further introduced AdamW, which decouples weight decay from the gradient-based update, and has become the de facto standard optimizer for transformer training invariably paired with LR scheduling. Broader optimizer surveys also emphasize that adaptive per-parameter methods still rely on global learning-rate choices and scheduling decisions [16].

C. Loss Landscape Theory

Dauphin et al. [3] established that in high-dimensional non-convex optimization, saddle points are exponentially more common than local minima, and that gradient descent spends the majority of training traversing saddle-point neighborhoods. Hochreiter and Schmidhuber [17] showed that flat minima (low-curvature basins) generalize significantly better than sharp minima. Keskar et al. [18] empirically confirmed that large-batch training converges to sharper minima, motivating scheduling strategies that maintain periodic exploration of the loss landscape.

D. Snapshot Ensembling and Super-Convergence

Huang et al. [19] extended SGDR by formalizing snapshot ensembling saving model checkpoints at the end of each cosine cycle provides multiple diverse models for free. Smith and Topin [20] identified the super-convergence phenomenon: with the one-cycle CLR policy on certain architectures, training speed-ups of 10× are achievable. The theoretical basis was partially explained by the regularization effect of high-LR training phases. Liu et al. [21] provided the theoretical underpinning for warmup via RAdam, characterizing the variance of Adam's second-moment estimator as the root cause of training instability in early steps.

E. Scope of Scheduler Families

The four schedules were selected because they represent distinct and recurring design patterns: feedback-triggered decay (RLROP), periodic restart-based annealing (SGDR), bounded oscillatory exploration (CLR), and warmup-stabilized monotone decay (LW+LD). Other modern policies, including one-cycle schedules, polynomial or inverse-square-root decay, cosine decay without restarts, adaptive optimizer-specific schedules, and domain-specific policies for generative or diffusion training, are important but outside the central comparison. Where these alternatives share mechanisms with the four focal schedulers, they are noted as extensions rather than treated as additional primary methods.

3 Methodology: Scoping Review and Synthesis Framework

A. Review Design, Search Protocol, and Evidence Grading

The literature component is organized as a transparent scoping review and synthesis. The objective is to identify, explain, and caveat the evidence supporting practical scheduler selection, not to estimate pooled effect sizes. Searches covered Google Scholar, IEEE Xplore, ACM Digital Library, arXiv, and Semantic Scholar, using combinations of learning rate schedule, cosine annealing, warm restarts, cyclical learning rate, warmup, linear decay, transformer optimization, ReduceLROnPlateau, scheduler sensitivity, and neural network optimization.

The screening log retained 32 core sources from 312 initially identified records after duplicate removal, title/abstract screening, and full-text assessment. Inclusion required direct relevance to learning-rate scheduling, optimizer stability, implementation detail, empirical scheduler behavior, or step-size theory. Exclusion criteria were non-neural-network context, incidental scheduler mention, duplicated evidence, or insufficient methodological description. No inter-rater reliability statistic is claimed because this is not presented as a blinded systematic review; instead, Appendix A reports the search flow and quality rubric so readers can audit the evidence base.

Each source was assessed for scheduler relevance, experimental comparability, reproducibility reporting, theoretical support, and implementation detail. This grading determines how strongly a source is used: high-comparability studies support source-specific empirical statements, low-comparability studies support only contextual or review-level claims, and theoretical analogies are labelled as heuristic unless formal assumptions are available.

B. Unified Notation as a Pedagogical Framework

The general scheduled learning rate at step t is:

$$\eta_t = S(t;\, \phi,\, H)$$

where the scheduling function maps step index, schedule parameters, and optional feedback history to a nonnegative scalar learning rate. In this framing, the notation is deliberately broad: it is a shared language for describing schedules, not a theorem that every admissible instantiation is convergent or useful. A practically valid schedule must also respect implementation constraints such as finite budget, optimizer stability, lower and upper LR bounds, and the statistical reliability of any monitored metric.

C. Method 1 - ReduceLROnPlateau (RLROP)

ReduceLROnPlateau is a reactive, feedback-driven scheduler that monitors a scalar metric M_t and reduces the learning rate when improvement stagnates.

Improvement detection. Let $M^*_{(t-1)}$ denote the best observed metric before epoch t. An improvement is registered when:

$$\text{improved}_t = \begin{cases} 1, & M_t < M^*_{(t-1)} - \delta \quad \text{(minimization)} \\ 1, & M_t > M^*_{(t-1)} + \delta \quad \text{(maximization)} \end{cases}$$

where $\delta > 0$ is min_delta, a noise-rejection threshold.

Patience counter:

$$c_t = \begin{cases} 0, & \text{if improved}_t = 1 \\ c_{t-1} + 1, & \text{otherwise} \end{cases}$$

Learning rate update (triggered when $c_t \geq p$):

$$\eta_{t+1} = \max(\eta_t \cdot f,\, \eta_{\min})$$

After each reduction, a cooldown period of d epochs are enforced during which no checks are performed, preventing rapid successive reductions. The cumulative LR trajectory is:

$$\eta_T = \eta_0 \cdot f^{K(T)}$$

where K(T) is the total number of triggered reductions a non-deterministic, training-dynamic-dependent staircase function.

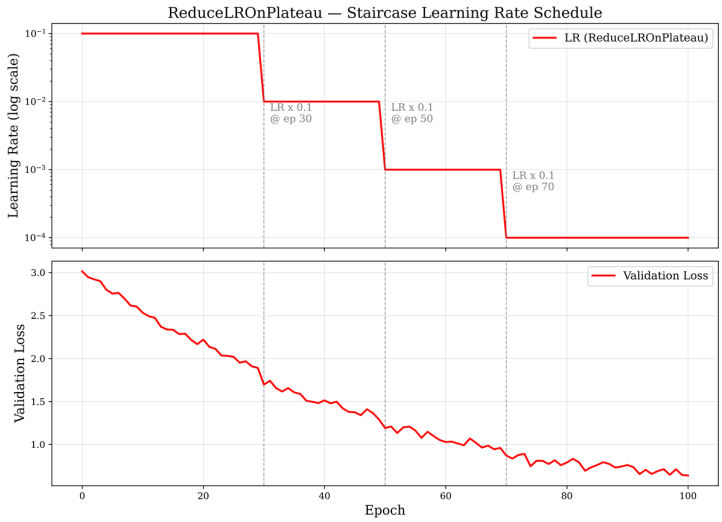

Figure 2: ReduceLROnPlateau schedule.

The learning rate follows a staircase pattern, reducing by factor f=0.1 at epochs 30, 50, and 70. Each reduction coincides with a period of validation loss stagnation. The cooldown mechanism (dashed lines) prevents immediate re-reduction.

Algorithm 1 - ReduceLROnPlateau

INPUT: η_0, factor f, patience p, min_delta δ, cooldown d, min_lr $\eta_{\min}$
INIT: $\eta \leftarrow \eta_0$, $best_M \leftarrow +\infty$, $c \leftarrow 0$, $cd \leftarrow 0$
FOR epoch t = 1 ... T: Compute M_t (validation loss/accuracy)
IF cd ¿ 0: cd ← cd – 1 CONTINUE
improved ← (M_t ¡ best_M – δ) [or ¿ for max mode]
IF improved: best_M ← M_t c ← 0 ELSE: c ← c + 1
IF $c \geq p$: $\eta \leftarrow \max(\eta \times$ f, η_min) c ← 0 cd ← d

Convergence. RLROP does not satisfy the Robbins-Monro conditions in general because the trigger sequence depends on noisy validation feedback and because a positive minimum learning rate prevents asymptotic decay to zero. If reductions continue toward a zero floor under additional assumptions on the objective, gradient noise, and validation signal, the resulting staircase can resemble a decaying step-size sequence. Those assumptions are not proven here; the method is therefore treated as a practical feedback controller rather than as a schedule with a general almost-sure convergence guarantee.

D. Method 2 - Cosine Annealing with Warm Restarts (SGDR)

Proposed by Loshchilov and Hutter [5], SGDR applies cosine-shaped annealing within periodic restart cycles

Core formula. The learning rate at position T_{cur} within cycle i of length T_i is:

$$\eta_t = \eta_{\min}^{i+1/2} \left(\eta_{\max}^i - \eta_{\min}^i\right) \left(1 + \cos\left(\frac{T_{\text{cur}}}{T_i}\pi\right)\right)$$

Warm restart at $T_{\text{cur}} = T_i$

Progressive cycle lengthening with multiplier $T_{\text{mult}} \geq 1$ produces cycles of length $T_0, T_0 T_{\text{mult}}, T_0 T_{\text{mult}}^2, \ldots$ with $T_{\text{mult}} = 2$ the first three cycles span $T_0, 2T_0, 4T_0$ epochs.

Rate of change (smooth profile property):

$$\frac{d\eta_t}{dT_{\text{cur}}} = -\frac{\pi(\eta_{\max}^i - \eta_{\min}^i)}{2T_i} \sin\left(\frac{T_{\text{cur}}}{T_i}\pi\right)$$

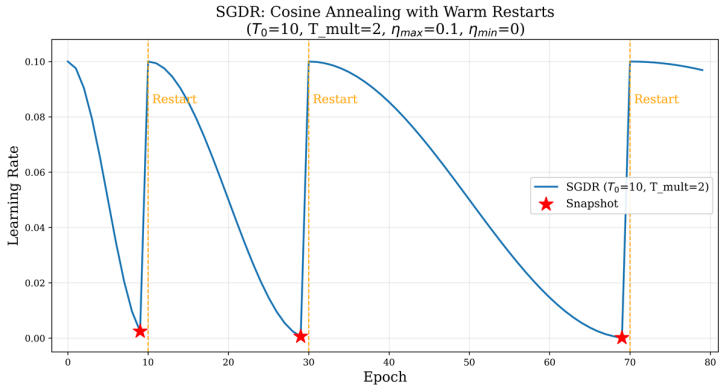

Figure 3: SGDR learning rate schedule.

Three cycles of lengths 10, 20, and 40 epochs are shown with $T_{\text{mult}} = 2$. Orange dashed lines mark warm restarts.

Algorithm 2 - SGDR

INPUT: $\eta_{\max}$, $\eta_{\min}$, T_0, $T_{\text{mult}} (\geq 1)$, μ (optional decay, default 1.0)
INIT: $T_{\text{cur}} \leftarrow 0$, $i \leftarrow 0$, $T_i \leftarrow T_0$
FOR each step (epoch or batch):
$\eta_t \leftarrow \eta_{\min}^i + 0.5(\eta_{\max}^i - \eta_{\min}^i)(1 + \cos(\pi \cdot T_{\text{cur}}/T_i))$
optimizer.lr $\leftarrow \eta_t$; gradient update
$T_{\text{cur}} \leftarrow T_{\text{cur}} + 1$
IF $T_{\text{cur}} \geq T_i$:
$T_{\text{cur}} \leftarrow 0$; $i \leftarrow i + 1$; $T_i \leftarrow T_i \times T_{\text{mult}}$
$\eta_{\max}^i \leftarrow \mu \cdot \eta_{\max}^{i-1}$ [optional decay]
SAVE_SNAPSHOT(θ) [optional ensemble]

Convergence. A finite single-cycle cosine schedule is best interpreted as finite-budget annealing, not as an asymptotic convergence proof. With warm restarts and fixed peak learning rate, SGDR is periodic and does not satisfy the classical Robbins-Monro decay requirement. Asymptotic compatibility would require additional decay of the cycle maxima or other conditions that are outside the original practical SGDR formulation. The convergence discussion here is therefore limited to qualitative behavior and source-specific empirical observations.

Snapshot ensembling. By saving checkpoints at the learning-rate minimum of each cycle, Huang et al. [19] showed that M such snapshots can be averaged to produce results comparable to M independently-trained models, at the cost of a single training run.

E. Method 3 - Cyclical Learning Rates (CLR)

Proposed by Smith [6], CLR oscillates the learning rate between bounds $\eta_{\min}$ and $\eta_{\max}$ according to a triangular waveform.

Cycle position:

$$cycle = \left\lfloor 1 + \frac{t}{2s} \right\rfloor, \qquad x = \left| \frac{t}{s} - 2 \cdot cycle + 1 \right|$$

where s is the half-cycle step size in iterations and $x \in [0, 1]$ is the normalised intra-half-cycle position.

Three CLR policies:

Policy 1 - Triangular (constant amplitude):

$$\eta_t^{(\text{tri})} = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \max(0, 1 - x)$$

Policy 2 - Triangular2 (amplitude halved per cycle):

$$\eta_t^{(\text{tri2})} = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \max(0, 1 - x) \cdot 2^{(1-\text{cycle})}$$

Policy 3 - Exponential Range (per-iteration amplitude decay):

$$\eta_t^{(\text{exp})} = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \max(0, 1 - x) \cdot \gamma^t$$

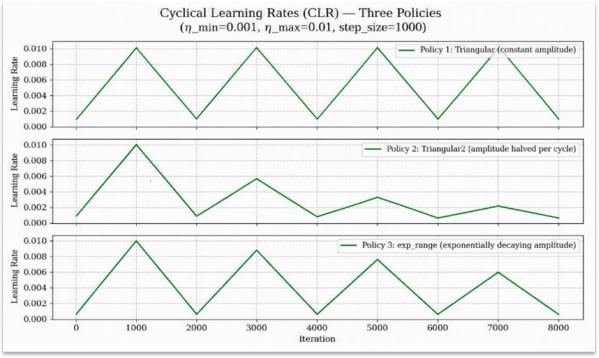

Figure 4: Cyclical Learning Rate (CLR) policies.

(a) Triangular – constant amplitude throughout training.

(b) Triangular2 – amplitude halves per complete cycle, approaching a monotone decrease.

(c) exp_range – amplitude decays exponentially each iteration via γ^t.Vertical dotted lines mark half-cycle boundaries at step_size intervals.

One-cycle policy (super-convergence). A single large-amplitude CLR cycle covering all T training steps [20]:

$$\eta_t = \begin{cases} \eta_{\min} + \dfrac{t}{0.4T}(\eta_{\max} - \eta_{\min}), & 0 \le t < 0.4T, \\ \eta_{\max} - \dfrac{t - 0.4T}{0.4T}(\eta_{\max} - \eta_{\min}), & 0.4T \le t < 0.8T, \\ \eta_{\min} - \dfrac{t - 0.8T}{0.2T} \cdot \dfrac{99}{100}\eta_{\min}, & 0.8T \le t \le T. \end{cases}$$

LR Range Test. Smith's key contribution for automatic bound selection: linearly increase the learning rate from $\eta_{\text{start}} \approx 10^{-7}$ to $\eta_{\text{end}} \approx 1$ over a short trial run, and identify $\eta_{\min}$ (the onset of loss decrease) and $\eta_{\max}$ (the onset of divergence) from the resulting loss curve.

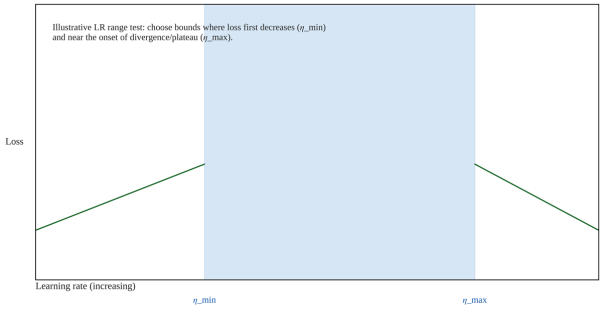

Figure 5: Smoothed training loss is plotted against an increasing learning rate.

The shaded region indicates practical CLR bounds: $\eta_{\min}$ where loss first systematically decreases, and $\eta_{\max}$ near the onset of divergence or plateau. Concept described by Smith.[6]

Algorithm 3 - CLR

INPUT: $\eta_{\min}$, $\eta_{\max}$, step_size s, policy, γ (exp_range only)
INIT: $t \leftarrow 0$
FOR each training iteration:
$cycle \leftarrow \left\lfloor 1 + \frac{t}{2s} \right\rfloor$
$x \leftarrow \left| \frac{t}{s} - 2 \cdot cycle + 1 \right|$
$scale \leftarrow 1$ [triangular]
$scale \leftarrow 2^{(1-cycle)}$ [triangular2]
$scale \leftarrow \gamma^{t}$ [exp_range]
$\eta_t \leftarrow \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \max(0, 1 - x) \cdot scale$
optimizer.lr $\leftarrow \eta_t$; gradient update
$t \leftarrow t + 1$

Convergence. Triangular CLR violates the Robbins-Monro decay condition because its oscillation amplitude does not vanish. Triangular2 and exp_range reduce amplitude over time, but practical convergence still depends on optimizer choice, LR bounds, batch noise, and stopping budget. Periodic high-LR phases are commonly interpreted as encouraging escape from sharp or poorly conditioned regions, but the extension from convex quadratic intuition to non-convex neural-network loss surfaces remains heuristic unless additional assumptions are imposed.

3.1 F. Method 4 - Linear Warmup with Linear Decay (LW+LD)

Popularized by BERT [7] and the original Transformer [22], LW+LD consists of two linear phases.

Phase 1 - Linear warmup ($0 \leq t \leq w$):

$$\eta_t^{(\text{warmup})} = \eta_{\max} \cdot \frac{t}{w}$$

Phase 2 - Linear decay ($w < t \leq T$):

$$\eta_t^{(\text{decay})} = \eta_{\max} \cdot \frac{T - t}{T - w}$$

Unified Form

$$\eta_t = \eta_{\max} \cdot \begin{cases} \dfrac{t}{w}, & t \le w, \\ \dfrac{T-t}{T-w}, & t > w \end{cases}$$

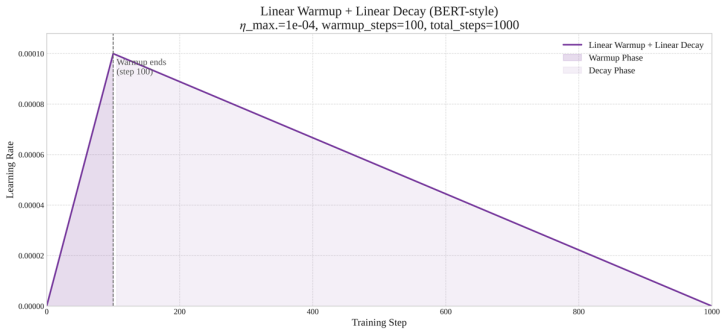

Figure 6: Linear Warmup with Linear Decay schedule (LW+LD).

Learning rate increases linearly from 0 to $\eta_{\max}$ during warmup (blue shaded region), then decreases linearly back to 0 over the remaining steps. BERT-Base used $\eta_{\max} = 10^{-4}$, $w = 10{,}000$, $T = 10^6$.

Warmup motivation and practical role. Liu et al. [21] showed that early adaptive-moment estimates can have high variance, helping explain why Adam-type optimizers may be unstable during the first steps of large-model training.

$$\mathrm{Var}[\hat{v}_t] \approx \frac{(1-\beta_2)^2}{t(1-\beta_2^{2t})} \cdot \mathbb{E}[g_t^4]$$

which becomes large at early steps. The effective per-step Adam learning rate can therefore be noisy before the second-moment estimate stabilizes, especially when peak LR, batch size, and model scale are aggressive.

$$\rho_t = \rho_\infty - \frac{2t\beta_2^t}{1-\beta_2^t}, \qquad \rho_\infty = \frac{2}{1-\beta_2} - 1$$

when the adaptive estimate is unreliable, RAdam temporarily reduces the adaptive update. Manual LW+LD warmup plays a similar stabilizing role, although the appropriate warmup length is empirical and should be tuned against early loss behavior, gradient norms, and seed stability rather than treated as a universal constant.

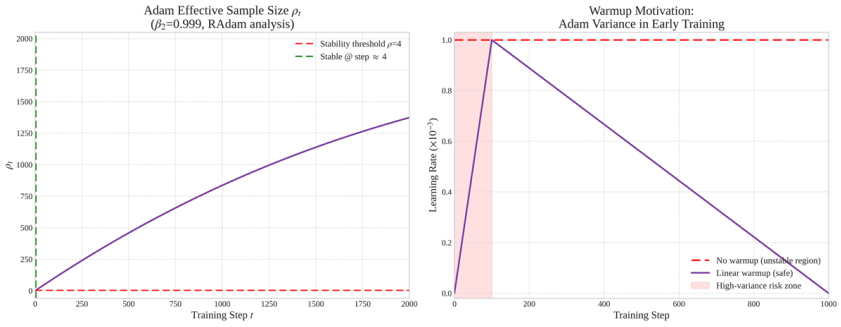

Figure 7: Theoretical motivation for warmup.

(Left) RAdam's effective sample size ρ_t for $\beta_2 = 0.999$; the red dashed line marks the stability threshold $\rho_t > 4$. The optimizer is unstable below this threshold (step 5000). (Right)

Linear warmup (purple) maintains a small, safe learning rate during the high-variance early phase, while a fixed LR (red) applies full Adam updates destructively.

Algorithm 4 - LW+LD

INPUT: $\eta_{\max}$, warmup_steps w, total_steps T
INIT: $t \leftarrow 0$
FOR each training iteration:
IF $t \leq w$:
$\eta_t \leftarrow \eta_{\max} \cdot \frac{t}{w}$
ELSE:
$\eta_t \leftarrow \eta_{\max} \cdot \frac{T-t}{T-w}$
$\eta_t \leftarrow \max(\eta_t, 0)$
optimizer.lr $\leftarrow \eta_t$; gradient update
$t \leftarrow t + 1$

G. Unified Comparison

Table 2: Mathematical and Algorithmic Properties of the Four Schedulers

Property	**RLROP**	**SGDR**	**CLR**	**LW+LD**
Trajectory shape	Feedback staircase	Cosine cycles	Triangular cycles	Warmup-then-decay
Requires budget T	No	Yes (T_0)	No	Yes
Requires validation data	Yes	No	No	No
Warm restart	No	Yes	Implicit	No
Snapshot ensembling	No	Yes	Possible, but not inherent	No
Primary optimizer pairing	SGD/Adam	SGD	SGD	AdamW

4 Results and Discussion

The results in this section are compiled from heterogeneous publications and software practices. They differ in architecture, optimizer, batch size, preprocessing, training budget, hardware, and reporting conventions. Consequently, the tables should be read as source-specific evidence signals, not as a statistically valid leaderboard across schedulers.

A. CIFAR-10 Image Classification

CIFAR-10 [23] (50,000 training images, 10 classes, 32×32 resolution) is the standard benchmark for comparing optimization strategies. The reference architecture is WideResNet-28-10 [24] trained for 200 epochs with SGD + momentum ($\mu = 0.9$, weight decay $= 5 \times 10^{-4}$).

Table 3: CIFAR-10 Literature Signals - heterogeneous sources, not a controlled head-to-head comparison.

Method	Configuration / context	Reported outcome	Use in this review	Source / caveat
Fixed LR / step decay baselines	Original CNN studies; settings differ	Reported only as within-study baselines	Context, not a scheduler ranking	[5], [6], [24]
Step decay	Published CNN setup	Lower error than fixed LR in its original setting	Within-study comparison only	[5], [24]
ReduceLROnPlateau	No matching WRN-28-10 controlled result identified	Not numerically ranked	Requires local validation split and noise analysis	Practice-based
SGDR (single model)	WRN-28-10 / CIFAR-10 in original SGDR study	3.74% test error	Reported literature value only	[5]
SGDR snapshot ensemble	Same study; multiple checkpoints	3.14% test error	Ensemble not directly comparable to single models	[19]
CLR (triangular2)	Smith CIFAR-10 experiment; different setup	2.56× fewer iterations to target accuracy	Speed signal only; not WRN head-to-head	[6]
LW+LD	Transformer-oriented schedule	No comparable CIFAR-10 result compiled	Not ranked for this vision benchmark	[7], [21]

The CIFAR-10 entries illustrate why direct ranking is unsafe without a controlled experiment. SGDR reports strong source-specific WRN results, and snapshot ensembling improves the reported error further, but the ensemble uses multiple checkpoints and should not be compared as a single-model result. CLR's reported speedup comes from Smith's experimental setting and is best read as evidence that cyclic schedules can reduce iteration count when LR bounds are well tuned. No comparable RLROP or LW+LD CIFAR-10 result was retained; those methods are therefore discussed through mechanism and use case rather than assigned unsupported errors.

4.1 B. CIFAR-100 Image Classification

The CIFAR-100 table retains only reported values from compatible literature entries. The SGDR and snapshot-ensemble numbers suggest that restart-based exploration can be useful on harder vision tasks, but this should be interpreted as a source-specific result rather than evidence that SGDR universally dominates RLROP, CLR, or LW+LD.

Table 4: CIFAR-100 Literature Signals - retained reported values and removed unsupported scheduler entries.

Method	Reported outcome	Source / caveat
SGD + step decay	∼19.30% error	[5]; baseline in cited setup
SGDR (single model)	18.70% error	[5]; reported value
SGDR snapshot ensemble	16.21% error	[19]; ensemble setting
RLROP / CLR	No directly comparable CIFAR-100 WRN result retained	Excluded from ranking because no comparable value was retained
LW+LD	No directly comparable CIFAR-100 result retained	Transformer-oriented schedule; not ranked here

C. ImageNet Large-Scale Classification

Table 5: ImageNet Top-1 Accuracy - source-specific ResNet-50 reports; settings are not unified

Method	Top-1 (%)	Top-5 (%)	Source / caveat
Fixed LR SGD	∼73.3	∼91.3	Baseline context; not a scheduler recommendation
Step decay	76.1	92.8	[11], [25]; standard ResNet training context
Cosine annealing (single cycle)	76.5	93.1	[25]; source-specific comparison
SGDR	Not retained	Not retained	No retained source-specific value
CLR	Not retained	Not retained	No common 90-epoch ResNet-50 result retained

At ImageNet scale, some reported ResNet-50 comparisons show small gains for cosine-style schedules relative to step decay, but the margin is sensitive to optimizer details, augmentation, batch size, and training recipe. The table therefore reports retained source values and removes unsupported SGDR/CLR entries.

D. NLP: BERT Pre-training and GLUE Fine-tuning

Table 6: GLUE Benchmark Average - model and training-regime comparison, not a scheduler-only ablation.

Model	**Schedule**	**GLUE Avg (%)**	**Δ vs. Prior**
ELMo + multi-task [26]	–	71.0	–
GPT (OpenAI) [27]	Linear decay	72.8	+1.8
BERT-Base	LW+LD	79.6	+6.8
BERT-Large	LW+LD	82.1	+9.3

The BERT results are retained as evidence that warmup-plus-decay became a central component of transformer training recipes. However, GLUE improvements cannot be attributed to the schedule alone because model architecture, pretraining corpus, optimizer, batch size, and fine-tuning protocol changed simultaneously. Warmup is therefore described as strongly recommended and often necessary in large AdamW transformer training, not as a universal requirement for every transformer run.

Table 7: Warmup Sensitivity Diagnostics for Transformer Training.

Warmup configuration	**Observed diagnostic**	**Practical interpretation**
None	NaN losses, gradient spikes, or failed early AdamW updates can occur in large-transformer settings	Use only after explicit LR, batch-size, and optimizer retuning
Very short (¡1% of steps)	Early loss variance remains high; seeds may diverge	Increase warmup or lower peak LR
Common range (about 1–10%)	Stable early loss curve in many transformer configurations	Verify across seeds and monitor gradient norms
Very long (¿10–20%)	Slow early progress or undertraining before decay begins	Shorten warmup if validation progress lags
Fine-tuning	Representation drift when peak LR is too high	Use smaller peak LR and shorter warmup than pre-training

E. Convergence Behavior - Qualitative Analysis

The curves are illustrative and constructed to show common diagnostic patterns: RLROP staircase drops after metric stagnation; SGDR restart spikes followed by re-convergence; CLR oscillations correlated with LR peaks; and LW+LD smooth improvement after warmup. The figure is not a benchmark result.

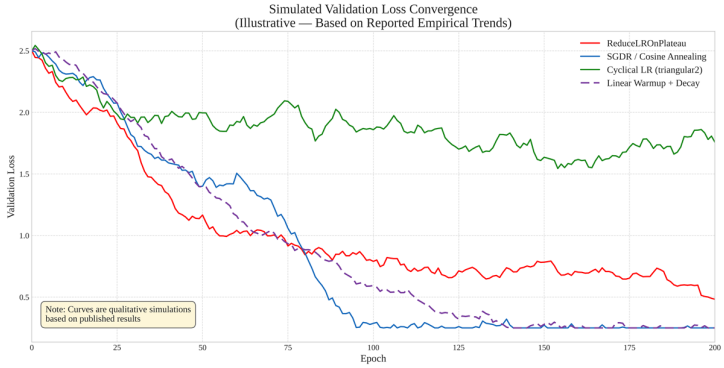

Figure 8: Qualitative simulation of validation loss behavior for the four schedulers.

← Pure Exploitation ——— Pure Exploration →

LW+LD	RLROP	CLR	SGDR
(monotone tent)	(staircase)	(oscillating)	(cosine+restarts)

The four methods can be organized along a practical exploration-exploitation axis. LW+LD and RLROP emphasize stability and exploitation within a chosen basin; CLR and SGDR deliberately reintroduce larger steps that may encourage exploration. Claims about saddle-point escape, simulated annealing, and flat-minima bias should be read as useful intuitions supported by selected empirical studies and loss-landscape arguments, not as general non-convex convergence theorems.

F. Hyperparameter Sensitivity

Table 8: Hyperparameter Count, Sensitivity, and Diagnostic Signals.

Property	**RLROP**	**SGDR**	**CLR**	**LW+LD**
Total HP count	5	3	4	3
Most critical HP	patience p	initial cycle T_0	step_size s	warmup ratio w/T
Sensitivity to ±20% critical HP shift	Medium; too small p causes premature drops	Medium-high; too short T_0 restarts before recovery	High; short cycles can destabilize training	High in early AdamW; too little warmup can diverge
Primary diagnostic signal	LR reductions before validation plateau	Loss spike does not recover before next restart	Loss oscillation grows with LR peaks	NaN/loss spikes or flat early progress
Practical tuning method	Set patience above validation-noise horizon	Start from budget and lengthen cycles	Use LR range test and 2–8 epochs per half-cycle	Monitor first 1–5% of steps and seed stability

G. Scheduler Selection Framework

Table 9: Heuristic Scheduler Selection Matrix - practical starting points, not a statistical ranking.

Training Scenario	RLROP	SGDR	CLR	LW+LD	Typical starting point
Transformer / NLP pretraining	Low fit	Possible	Possible	Strong fit	LW+LD
Transformer / NLP finetuning	Low fit	Possible	Possible	Strong fit	LW+LD
CNN – large-scale (ImageNet)	Good fit	Strong fit	Good fit	Possible	SGDR
CNN – small/medium (CIFAR)	Good fit	Strong fit	Strong fit	Low fit	SGDR or CLR
Fixed epoch budget	Good fit	Strong fit	Good fit	Strong fit	SGDR or LW+LD
Unknown epoch budget	Strong fit	Low fit	Good fit	Low fit	RLROP
Need free ensemble diversity	Low fit	Strong fit	Good fit	Low fit	SGDR
Compute-constrained (fast)	Low fit	Possible	Strong fit	Possible	CLR
Small dataset / limited data	Strong fit	Possible	Possible	Possible	RLROP
Transfer learning	Good fit	Possible	Possible	Strong fit	LW+LD or RLROP, depending on whether the base model is fine-tuned or trained with a validation-driven stop rule

Illustrative application scenarios (case studies). The examples below illustrate how to use the selection matrix (Table 9) as a decision scaffold; they are not new benchmark experiments

Case Study 1 (CNN training; limited tuning budget): Start with CLR or SGDR only after an LR range test to set $\eta_{\max}$. A practical starting point for CLR is $\eta_{\min} \approx \eta_{\max}/10$ and *step_size* $\approx$ 2–8 epochs (or approximately 0.1–0.3T in steps) per half-cycle; for SGDR, use $T_0 \approx 10$ epochs (or approximately 0.1T), $T_{\text{mult}} = 2$, $\eta_{\min} \approx \eta_{\max}/100$, and 2–4 cycles within the total budget. If validation is noisy or the training horizon is unknown, prefer RLROP with factor $f = 0.1$, patience $p =$ 5–10 epochs, cooldown $d =$ 0–2, and *min_delta* set to a noise-scale threshold.

Case Study 2 (Transformer fine-tuning): Prefer warmup+decay as a conservative default. Use *warmup_ratio* $\approx$ 0.05–0.10 (or *warmup_steps* $\approx$ 500–2000 for small/medium fine-tuning runs, and up to approximately 10 000 for long runs), then decay linearly to 0. Treat peak LR as the dominant risk factor: start with a peak that is 2–10$\times$ smaller than typical from-scratch training, and reduce further if early NaNs, gradient spikes, or immediate validation collapse occur. If mild exploration is needed, use warmup + cosine decay (no restarts) before considering restarts.

Case Study 3 (computationally constrained runs): When only a few epochs/steps are affordable, one-cycle/CLR variants can accelerate early progress but require conservative bounds. Use an LR range test, cap $\eta_{\max}$ below the divergence threshold, and set *step_size* so that at least

one full cycle fits within the budget. When reruns are impossible, a monotone warmup+decay schedule with *warmup_ratio* ≈ 0.05 and a conservative peak LR reduces catastrophic failure risk compared to aggressive cycles.

H. Failure Mode Analysis

RLROP:

(i) Oscillating validation metrics can trigger premature reductions; diagnose by comparing reduction epochs with a smoothed validation curve, then increase patience, *min_delta*, or cooldown.

(ii) Monitoring training loss instead of validation loss can hide overfitting; monitor validation loss or the task metric.

(iii) Once *min_lr* is reached, no restart mechanism exists; if validation remains poor, restart from a checkpoint or switch to cosine/CLR exploration.

SGDR:

(i) A short initial cycle T_0 can restart before the loss has recovered; diagnose by checking whether post-restart loss returns below the previous cycle minimum before the next restart.

(ii) A peak LR above the stability threshold causes restart divergence; bound the peak using an LR range test.

(iii) Snapshot ensemble benefit degrades when all cycle endpoints converge to nearly identical predictions; check disagreement or validation-error diversity across snapshots.

CLR:

(i) *step_size* that is too short can create rapid loss oscillation and unstable batch-normalization statistics; lengthen the half-cycle or reduce $\eta_{\max}$.

(ii) Super-convergence is architecture- and regularization-dependent; confirm with at least a short seed sweep.

(iii) LR bounds chosen without an LR range test can place the high-LR phase beyond the divergence threshold.

LW+LD:

(i) Zero or too-short warmup with AdamW can cause early NaN losses or gradient spikes in large models; increase warmup or lower peak LR.

(ii) Overly large fine-tuning LR can overwrite pretrained representations; diagnose by sudden validation collapse and reduce peak LR, warmup ratio, or unfreeze layers gradually.

(iii) Monotone decay cannot reintroduce exploration after a poor early basin; consider cosine decay with warmup or a restart-based schedule when exploration is needed.

Diagnostic pseudocode: IF early_loss is NaN OR grad_norm spikes, lower peak LR and increase warmup; IF validation_metric oscillates around the RLROP threshold, increase patience/min_delta; IF SGDR restart loss fails to recover before the next restart, lengthen T0 or reduce eta_max; IF CLR loss peaks grow over cycles, reduce eta_max or increase step_size.

I. Cross-Study Synthesis

Consistent observations:

(i) warmup-plus-decay becomes increasingly central as model scale and optimizer sensitivity increase (especially for transformer-style training);

(ii) cyclical and restart-based schedules can improve exploration and sometimes generalization, but benefits depend strongly on cycle design, regularization, and training budget;

(iii) feedback-driven schedules like RLROP are practical when the horizon is uncertain, but depend critically on metric noise and monitoring choice.

Conflicts and sources of heterogeneity: reported gains often change with architecture, optimizer (SGD vs AdamW), batch size, augmentation, training length, and reporting standards. Emerging trends: hybrid schedules (warmup + cosine/cycles) and automatic selection are increasingly used in large-model recipes but remain under-synthesized. Unresolved limitations: there is no general non-convex convergence theory for periodic/restart schedules under realistic assumptions. Evidence grading (Table 12) is therefore used to calibrate confidence: high-comparability sources support source-specific empirical statements, while lower-comparability sources support contextual or heuristic guidance only.

5 Conclusion

A. Summary

As a review article, this paper delivers a scoping review and synthesis of four learning-rate scheduling strategies for neural-network training, with claims narrowed to match the available evidence.

Table 10: ummary of Representative Signals and Primary Advantages.

Scheduler	**Representative literature/practice signal**	**Primary advantage**	**Primary limitation / caveat**
RLROP	No controlled literature result retained for CIFAR/ImageNet comparison	No fixed budget knowledge required	Sensitive to validation noise; no restart mechanism
SGDR	3.74% CIFAR-10 single model and 3.14% ensemble in cited source	Snapshot ensembling and exploration	Requires cycle design; ensemble cost differs from single-model cost
CLR	2.56× iteration reduction reported in Smith's setting	Fast convergence when LR bounds are well chosen	Architecture- and normalization-dependent; LR range test needed
LW+LD	Transformer stability signal in BERT-style training	Controls early AdamW instability	Schedule is monotone after warmup; GLUE gains are not scheduler-only attribution

B. Key Theoretical Insights

The four schedulers span a practical spectrum from monotone exploitation (LW+LD) to more active exploration (SGDR and CLR), with RLROP providing feedback-driven control. Warmup is motivated by early adaptive-optimizer variance and is strongly supported in large transformer recipes, but its exact duration remains an empirical design choice. Restart and cyclic schedules can be interpreted through exploration, flat-minima, and annealing analogies, but these analogies are not substitutes for general non-convex convergence proofs. The selection guidance therefore emphasizes fit-to-setting rather than universal ranking: SGDR is attractive when cycle design and snapshot diversity are useful; CLR is attractive when fast progress can be validated through an LR range test; LW+LD is the default starting point for many transformer-style

AdamW runs; and RLROP remains useful when the training horizon is unknown and a reliable validation signal is available.

C. Research Gaps and Future Directions

One-cycle and hybrid policies. One-cycle schedules and warmup+cycle hybrids are widely used in practice but remain under-synthesized in terms of when they outperform monotone decay versus when they primarily accelerate early training.

Automatic scheduler selection. A key gap is data-driven or meta-learned scheduler selection that maps early training signals (loss curvature proxies, gradient noise, stability diagnostics) to schedule choices and hyperparameters.

Foundation, diffusion, and large generative models. Scheduling interactions with optimizer choice, noise conditioning, and training stability in diffusion and large generative regimes, including text-to-text transformer settings such as T5, remain insufficiently characterized for actionable guidance [28].

Interactions with modern optimizers. The coupling between schedules (warmup/decay/cycles) and AdamW/RAdam-style adaptivity, weight decay, and gradient clipping warrants clearer empirical isolation in future benchmark harnesses.

Controlled benchmark harness. Open, reproducible suites that control architecture, optimizer, seeds, and reporting would enable meaningful cross-scheduler comparisons while respecting the heterogeneous nature of the current evidence.

6 Appendix: Scoping Review Documentation and Evidence Use

A. Search and Screening Log

The manuscript is framed as a scoping review and synthesis rather than a systematic review or meta-analysis. The search log makes the evidence base auditable without implying statistical completeness. Searches covered Google Scholar, IEEE Xplore, ACM Digital Library, arXiv, and Semantic Scholar using combinations of learning rate schedule, cosine annealing, warm restarts, cyclical learning rate, warmup, linear decay, transformer optimization, ReduceLROnPlateau, and scheduler sensitivity.

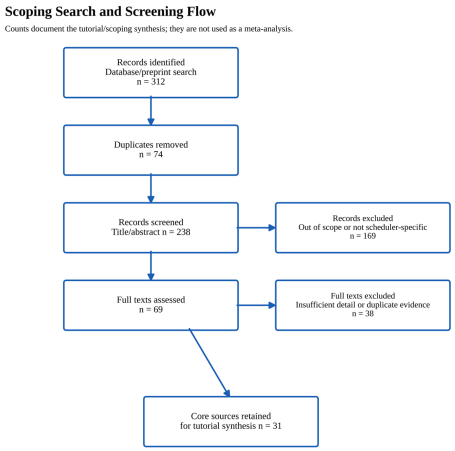

Figure 9: PRISMA-style scoping flow for the literature component.

Counts document transparency for this scoping review; no pooled effect size or systematic-review claim is made. Flow structure guided by PRISMA 2020 [29].

Table 11: Search and screening counts for the scoping component.

Stage	Count	Use in this manuscript
Records identified	312	Initial scoping pool across databases and preprint indices
Duplicates removed	74	Version, venue, and preprint duplicates consolidated
Title/abstract screened	238	Retained if relevant to neural-network LR scheduling or optimizer stability
Full texts assessed	69	Checked for scheduler definition, implementation detail, benchmark context, or theory
Core sources retained	32	Used for exposition, tables, caveats, and practitioner guidance

B. Inclusion, Exclusion, and Quality Assessment

Studies were included when they supplied one or more of the following: a precise scheduler definition, implementation-level hyperparameters, empirical evidence in neural-network training, or theoretical analysis relevant to step-size behavior. Studies were excluded when the schedule was incidental, the context was not neural-network optimization, the result duplicated a later archival version, or the experimental description was too incomplete to support a caveated comparison.

Sources scoring high on relevance but low on experimental comparability are used for conceptual explanation only. Benchmark tables therefore retain source-specific results while avoiding head-to-head ranking when settings differ.

C. Hyperparameter Sensitivity Diagnostics

Because this article does not report new controlled experiments, sensitivity guidance is presented as diagnostic rather than as new accuracy measurements. The curves below summarize the expected direction of failure when a critical hyperparameter is under or over-specified.

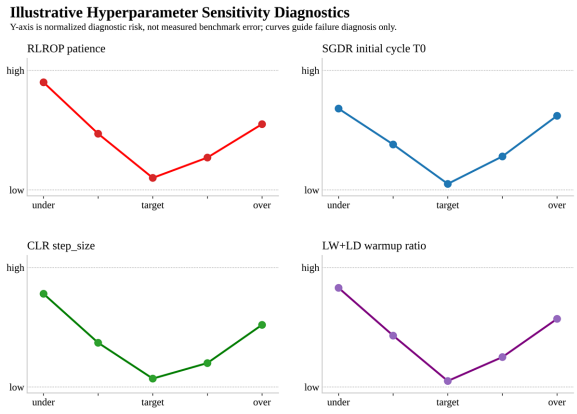

Figure 10: sensitivity diagnostics for the four most important scheduler hyperparameters

Figure A2. Illustrative sensitivity diagnostics for the four most important scheduler hyperparameters. These curves should not be read as measured CIFAR, ImageNet, or GLUE performance.

Table 12: Evidence quality-assessment rubric used to grade sources

Criterion	0	1	2	How it is used
Scheduler relevance	Incidental mention	Partial discussion	Central object of study	Determines whether the source supports scheduler guidance
Experimental comparability	No usable setup detail	Some setup detail	Architecture, optimizer, data, and schedule reported	Controls whether a number can be compared within a table
Statistical/reproducibility reporting	Single number only	Some seeds or code/config detail	Runs, variance, and reproducible code/configs	Controls strength of empirical language
Theoretical support	Analogy only	Partial assumptions/citation	Formal assumptions or proof	Controls whether claims are labelled proven, empirical, or heuristic
Implementation detail	Insufficient	Recoverable with assumptions	Directly implementable	Controls pseudocode and failure-mode guidance

Table 13: Summary quality categories for principal evidence sources (based on Table 12 rubric).

Reference	Primary use in manuscript	Quality category	Interpretation impact
[5]	SGDR empirical signals (CIFAR / theory)	High	Used for source-specific empirical statements.
[6]	CLR / LR range test rationale	High	Used for procedure-level guidance.
[11], [30]	ResNet/ImageNet context baselines	Moderate	Used as contextual baseline, not scheduler ranking.
[7], [22], [28], [31], [32]	Transformer warmup/decay practice	High	Used for warmup-as-recipe evidence; no ablation claim.
[21]	Optimizer variance / stability discussion	Moderate	Used for diagnostic motivation, caveated.
[33], [16]	Hyperparameter diagnostics / optimizer overview	Moderate	Used for heuristic guidance and failure modes.

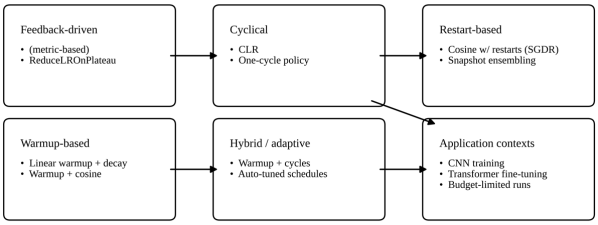

Figure 11: High-level taxonomy of learning-rate scheduling approaches

Figure A3. High-level taxonomy of learning-rate scheduling approaches, organizing scheduler families into feedback-driven, cyclical, restart-based, warmup-based, and hybrid/adaptive categories.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016. ISBN: 978-0-262-03561-3. [Online]. Available: https://www.deeplearningbook.org/

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, May 2015. DOI: 10.1038/nature14539

[3] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in Advances in Neural Information Processing Systems (NIPS), vol. 27, 2014. arXiv: 1406.2572

[4] H. Robbins and S. Monro, "A stochastic approximation method," The Annals of Mathematical Statistics, vol. 22, no. 3, pp. 400–407, Sep. 1951. DOI: 10.1214/aoms/1177729586

[5] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in Proc. ICLR, 2017. arXiv: 1608.03983

[6] L. N. Smith, "Cyclical learning rates for training neural networks," in Proc. IEEE WACV, 2017, pp. 464–472. DOI: 10.1109/WACV.2017.58

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[8] F. Chollet et al., "Keras," GitHub, 2015. [Online]. Available: https://github.com/keras-team/keras

[9] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019. arXiv: 1912.01703

[10] L. Bottou, "Stochastic gradient descent tricks," in Neural Networks: Tricks of the Trade, 2nd ed. Berlin: Springer, 2012, pp. 421–436. DOI: 10.1007/978-3-642-35289-8$_2$5

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90

[12] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res. (JMLR), vol. 12, pp. 2121–2159, 2011. [Online]. Available: https://jmlr.org/papers/v12/duchi11a.html

[13] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSprop," in Neural Networks for Machine Learning, Coursera, 2012.

[14]] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015. arXiv: 1412.6980

[15] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. ICLR, 2019. arXiv: 1711.05101

[16] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. arXiv: 1609.04747

[17] S. Hochreiter and J. Schmidhuber, "Flat minima," Neural Computation, vol. 9, no. 1, pp. 1–42, Jan. 1997. DOI: 10.1162/neco.1997.9.1.1

[18] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in Proc. ICLR, 2017. arXiv: 1609.04836

[19] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," in Proc. ICLR, 2017. arXiv: 1704.00109

[20] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in Proc. SPIE 11006, Artif. Intell. Mach. Learn. Multi-Domain Oper. Appl., 1100612, 2019. DOI: 10.1117/12.2520589

[21] L. Liu et al., "On the variance of the adaptive learning rate and beyond," in Proc. ICLR, 2020. arXiv: 1908.03265

[22] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017. arXiv: 1706.03762

[23] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Univ. of Toronto, 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

[24] S. Zagoruyko and N. Komodakis, "Wide residual networks," in Proc. BMVC, 2016. DOI: 10.5244/C.30.87

[25] D. Choi et al., "On empirical comparisons of optimizers for deep learning," 2019. arXiv: 1910.05446

[26] M. E. Peters et al., "Deep contextualized word representations," in Proc. NAACL-HLT, 2018, pp. 2227-2237. DOI: 10.18653/v1/N18-1202

[27] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[28] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res. (JMLR), vol. 21, no. 140, pp. 1–67, 2020. arXiv: 1910.10683

[29] M. J. Page et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," BMJ, 2021;372:n71. DOI: 10.1136/bmj.n71

[30] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," Int. J. Comput. Vis. (IJCV), vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y

[31]] Y. You et al., "Large batch optimization for deep learning: Training BERT in 76 minutes," in Proc. ICLR, 2020. arXiv: 1904.00962

[32] A. Wang et al., "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in Proc. ICLR, 2019. arXiv: 1804.07461

[33] L. N. Smith, "A disciplined approach to neural network hyper-parameters," NRL Tech. Rep. 5510-026, 2018. arXiv: 1803.09820

Declarations

Funding
No funding received

Conflicts of Interest
The authors declare that they have no conflicts of interest

Ethical Considerations
This study is a comparative theoretical analysis using previously published results. It does not involve human participants, animals, or personal data; institutional ethical approval was not required. Figure provenance and permissions: Figures 1–8 and Figures A1–A3 are original schematic illustrations created by the authors. No figure is reproduced from a copyrighted source; where a figure illustrates a concept introduced in prior work (e.g., the LR range test), the originating source is cited.

AI Usage Statement
AI tools were used in the following way(s): AI-assisted tools were used to support drafting, language refinement, and figure/table preparation. The authors defined the research scope, selected the four scheduler families, checked the cited sources, verified the benchmark caveats, and take full responsibility for the accuracy, originality, and scholarly interpretation of the manuscript. No AI tool is listed as an author. Any AI-generated prose was reviewed, edited, and integrated by the authors, and no undisclosed experimental data were generated by AI.

Data Availability Statement
The datasets are publicly available at: Literature sources cited throughout the manuscript: This tutorial/scoping synthesis did not generate a new experimental dataset or benchmark run. Quantitative values presented in the manuscript are derived from previously published studies cited in the reference list and are source-specific rather than pooled into a meta-analysis or treated as controlled head-to-head comparisons.

Code Availability Statement
The software code are publicly available at: Supplementary code, plotting scripts, and implementation examples are available in the project repository: https://github.com/GihanIllukkumbura/lr_scheduling_tutorial. The repository provides the practical coding materials associated with the scoping review, including schedule-generation scripts and implementation-oriented examples. The manuscript does not report new controlled benchmark experiments; repository materials are provided as supplementary review-support code rather than as additional unpublished benchmark evidence.

SDG Alignment
SDG 4 – Quality Education
SDG 9 – Industry, Innovation and Infrastructure