CAN CONTROL LEARNING WITH BANDIT OPTIMIZATION ACCELERATE CONVERGENCE IN MULTIMODAL FUSION OF MEDICAL DATA?

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal fusion has become a central paradigm in medical AI, integrating heterogeneous sources such as imaging and multi-omics for diverse downstream tasks. However, its training-time efficiency remains underexplored. As training progresses, modalities and tasks contribute unevenly to learning, yet prior methods often optimize the shared multi-task loss using fixed or heuristic open-loop weighting. This mismatch limits adaptation to evolving learning dynamics and can delay convergence to strong validation performance. To address this challenge, we propose **B**andit with **C**ontrol-**L**earning **A**ware optimizatio**N** (BCLAN), a novel *parameter-free* hybrid closed-loop controller that augments standard optimizers without architectural changes or additional trainable backbone parameters. Driven by validation feedback, BCLAN couples adversarial bandit exploration over a compact set of loss-weight templates with proportional–integral–derivative (PID)-based refinement in logit space, enabling it to *search* for improved task–modality allocations while *stabilizing* their adaptation under noisy, non-stationary feedback. We provide theoretical insights into this coupling, establishing bounded loss-weight refinement and identifying conditions for faster convergence. Across twelve heterogeneous-modality benchmarks in the medical domain, BCLAN improves downstream performance by up to $\approx 2.7\%$ and reduces training time by up to $\approx 50.4\%$ compared with the corresponding baselines, achieving a favorable training-time efficiency–performance trade-off. Code - BCLAN

1 INTRODUCTION

Multimodal fusion is crucial for medical AI, where heterogeneous sources, such as imaging, omics, and *electronic health records* (EHR), provide complementary evidence for diverse downstream tasks (e.g., multi-disease analysis, patient survival, and mortality prediction) (Hemker et al., 2024; Dhar et al., 2026c; Hemker et al., 2025). By exploiting such modalities, such fusion frameworks often outperform unimodal baselines (Dhar et al., 2026c; Hemker et al., 2025). Hence, there is growing interest in multimodal systems for multi-task prediction in medical AI. Recent advances (Hemker et al., 2024; Dhar et al., 2026c; Lai et al., 2024) prioritize cross-modal interactions, pursuing efficiency gains (e.g., fewer parameters and FLOPs) mainly through architectural design.

Much less attention has been devoted to the problem of *training-time efficiency*—i.e., how quickly multimodal fusion pipelines converge to a validation regime that guarantees good generalization (Cueto-Mendoza & Kelleher, 2024). In multi-task learning, the learning objective is usually a combination of task-specific losses that are combined into a loss mixture (Dhar et al., 2025). For multimodal fusion, where the optimization dynamics are often non-stationary, as shared representations evolve, the effective signal-to-noise ratio, gradient scales, and optimization difficulty can shift unevenly across modalities (Wang et al., 2020; Yang et al., 2024; Wei et al., 2025). When the multi-task objective is defined in an open-loop manner (e.g., using fixed weights for the different loss components), the loss allocation cannot track these shifts, which may delay stabilization of the validation regime loss (see Fig. 1) (Kendall et al., 2018; Chen et al., 2018; Wang et al., 2020). This mismatch yields uneven generalization dynamics, since modalities may overfit at different rates, and dominant modalities can suppress weaker ones via shared fusion parameters (Nowdeh et al., 2025). This slows convergence and delays entry into a desirable validation regime (see App. F.1). In this work, we hypothesize that *closed-loop optimization can accelerate the training convergence of multimodal fusion systems, thereby improving the training time efficiency–performance trade-off of the latter.*

We investigate this hypothesis by considering how to make the optimization loop *adaptive*, using loss-driven feedback to update the weights of different loss components as optimization progresses

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

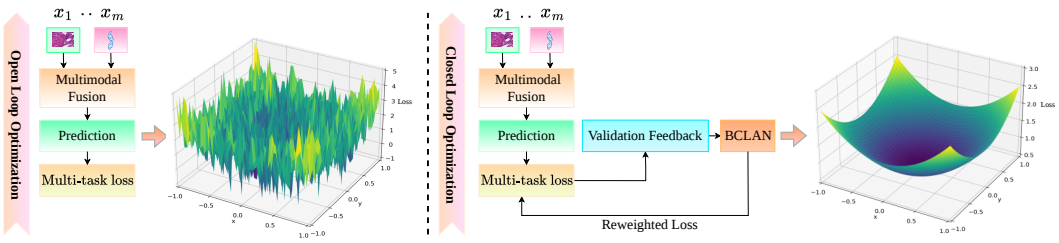

Figure 1: **Open-loop versus closed-loop optimization in multimodal fusion.** Conventional multimodal fusion (e.g., `MAIL` (Dhar et al., 2026c) and `CURE` (Dhar et al., 2026b)) optimizes a fixed multi-task loss in an open loop (*left*), so task–modality weighting cannot adapt to validation feedback as optimization dynamics drift during training. `BCLAN` closes this loop (*right*) by feeding validation feedback to the optimizer and adaptively reweighting the loss through bandit-guided search and `PID`-stabilized refinement (see §. 2.3). Schematic 3D loss landscapes show optimization behavior: compared with open-loop training, closed-loop adaptation steers optimization toward broader, flatter low-loss basins, leading to faster convergence without pipeline changes.

(see Fig. 1). Bandit-based schemes (Ganapathiraman et al., 2023; Liu et al., 2020) can be used to operationalize this principle through an explicit explore–exploit strategy that adapts sampling or update policies from loss feedback to produce loss weights that accelerate convergence. While these mechanisms have been shown successful for unimodal pipelines, it is less clear that they will remain stable under heterogeneous feedback. Most notably, directly applying bandit updates to regulate a multi-task learning loss in multimodal fusion can lead to high-variance updates (Kocák et al., 2014; Pogodin & Lattimore, 2020) under noisy and non-stationary validation regimes (Besbes et al., 2014; Resler & Mansour, 2019). This can trigger abrupt weight oscillations (Sener & Koltun, 2018), potential instability (Javaloy & Valera, 2021), and/or collapse into a subset of active terms, consequently impeding convergence (Peng et al., 2022; Zhang et al., 2024) (see Fig. 4).

To stabilize adaptation, we consider classical proportional–integral–derivative (`PID`) control, which regulates optimization trajectories through tracking-error-driven *control inputs* (Åström & Murray, 2008). However, these controllers lack explicit exploration, which limits their ability to track drifting optima as the effective objective evolves. As a result, either approach alone can fall short of **fast, reliable convergence**, thereby limiting the **training-time efficiency–performance trade-off** (see Figs. 2 and 4, Table 1 and Figs. 10–11 in App. F.1).

To address the challenge of designing a principled closed-loop optimization method that can jointly *search* for better allocations and remain *stable* under heterogeneous feedback, we propose a hybrid closed-loop *loss-reweighting* mechanism, **B**andit with **C**ontrol-**L**earning **A**ware optimizatio**N** (`BCLAN`). `BCLAN` is *parameter-free* and adaptively recalibrates the components of a multi-task objective at a coarse timescale (e.g., once per validation interval). It integrates two complementary online mechanisms: (i) an adversarial bandit component that performs coarse-grained explore–exploit search over a small set of loss-weight templates, and (ii) `PID`-based loss weight refinement driven by validation-loss feedback, to stabilize and fine-tune the selected loss mixture. This design makes `BCLAN` applicable to existing multimodal fusion pipelines without any architectural modification, while simply augmenting a standard base optimizer (e.g., `Adam`/`SGD` (Kingma & Ba, 2014; Ruder, 2016)). These components enable faster and more reliable convergence when learning shared representations from heterogeneous modalities.

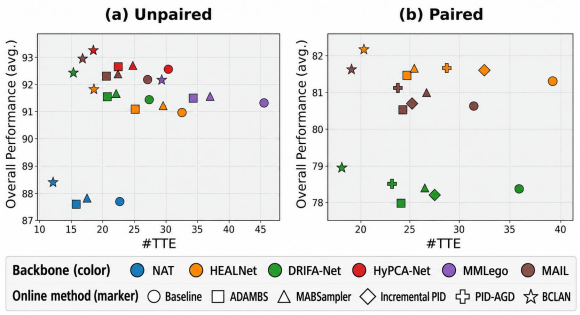

Figure 2: **Training time efficiency–performance trade-offs on heterogeneous benchmarks.** Unpaired (a) and paired (b) regimes. Points show backbone–method pairs from Tables 1–2; colors denote backbones and markers denote online-adaptation methods.

Our experimental results demonstrate the effectiveness of `BCLAN` on multimodal fusion, as shown in Fig. 2 and Tables 1–3. In particular, our method exhibits faster and reliable convergence, highlighting better task–modality allocations while keeping stable convergence (Figs. 2 and 4), indicating improved downstream performance gains. The main contributions of our work are:

1. We introduce BCLAN, a *new state-of-the-art hybrid closed-loop optimization controller* for multimodal fusion, enabling faster convergence and an improved training-time efficiency–performance trade-off (**Tables 1–3**), making it well-suited for resource-constrained medical AI applications.
2. We provide insights into BCLAN's effects, showing how bandit-driven explore–exploit allocation over loss-weight templates coupled with PID-based stabilization yields stable and bounded loss-weight adaptation, thereby promoting faster and reliable convergence (**Fig. 4 and Figs. 10–11**).
3. We conduct rigorous evaluations on twelve paired and unpaired heterogeneous-modality aware medical benchmarks, demonstrating that BCLAN achieves a favorable balance between training-time efficiency and downstream performance over existing baselines (see **Tables 1–3**).

Related work. We include an extensive discussion of related work in Appendix A.

2 METHOD

2.1 PRELIMINARIES

Let $\mathcal{X} = \{x_i\}_{i=1}^m$ denote m heterogeneous modalities in medical domain, with imaging tensors $x_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ (height, width, channels) or non-imaging vectors $x_i \in \mathbb{R}^{D_i}$ (e.g., multi-omics and EHR). The latter are rasterized into $H_i \times W_i \times C_i$ pseudo-images for unified convolutional processing. Given labels $\mathcal{Y} = \{y_t\}_{t=1}^T$ for T tasks, a fusion model with parameters β learns shared representations $\{x_i^S\}_{i=1}^m$ and predicts $\hat{\mathbf{y}} = \mathcal{F}(\{x_i^S\}_{i=1}^m; \beta)$. Let $\mathcal{L}_t^i(\beta)$ denote the loss for task t and modality i. The multi-task learning (MTL) objective is

$$\mathcal{L}_{\texttt{MTL}}(\beta; \Lambda) = \sum_{t=1}^{T} \sum_{i=1}^{m} \lambda_t^i \, \mathcal{L}_t^i(\beta), \qquad \Lambda = \{\lambda_t^i\}_{t,i} \in \Delta_T^m, \tag{1}$$

where $\Delta_T^m = \{\Lambda : \lambda_t^i \geq 0, \, \sum_{t,i} \lambda_t^i = 1\}$ is the simplex over the Tm loss components. The weights determine their relative contributions to shared parameter updates.

2.2 PROBLEM FORMULATION

Given the multimodal fusion framework in §. 2.1, we aim to *accelerate convergence* by adapting the task–modality weights $\Lambda_e \in \Delta_T^m$ of the MTL objective in Eq. 1, without architectural modification. In multimodal learning, task difficulty, gradient statistics, and modality-specific learning progress can evolve as shared representations develop (Islam & Iqbal, 2022; Wang et al., 2020; Chen et al., 2018). Static, open-loop weighting cannot track these changes, potentially delaying validation-loss reduction and prolonging training. Earlier convergence may also support stronger downstream performance under finite budgets, motivating *validation-driven loss allocation* to improve the *training-time efficiency–performance trade-off*.

Validation-loss trajectory and convergence-time objective. Let β_e denote the model parameters after epoch e and $\mathcal{E}_{\text{val}} \subseteq \{1, 2, \ldots\}$ the set of validation epochs. For $e \in \mathcal{E}_{\text{val}}$, define

$$\mathcal{J}_e \triangleq \mathcal{L}_{\texttt{MTL}}\Big(\beta_e; \Lambda^{\text{ref}}\Big), \tag{2}$$

where the objective is evaluated on validation data under fixed uniform reference weights $\Lambda^{\text{ref}} \in \Delta_T^m$. This keeps losses comparable across epochs despite adaptive training weights. At these epochs, the best-so-far validation loss is

$$\bar{\mathcal{J}}_e \triangleq \min_{\substack{k \in \mathcal{E}_{\text{val}} \\ k \leq e}} \mathcal{J}_k. \tag{3}$$

Given a validation-loss threshold $\mathcal{J}^\star$ and measured per-epoch training times $\tau_e > 0$ of epoch e, the *validation-loss convergence time* is

$$\mathcal{T}^\star \triangleq \min_{\substack{e \in \mathcal{E}_{\text{val}} \\ \bar{\mathcal{J}}_e \leq \mathcal{J}^\star}} \sum_{k=1}^{e} \tau_k, \tag{4}$$

which measures cumulative training time until the specified criterion is first satisfied, with $\mathcal{T}^\star = \infty$ if it is never met. For a given fusion backbone and base optimizer, let π denote an online loss-allocation rule selecting $\Lambda_e \in \Delta_T^m$ using only feedback available before epoch e. The objective is

$$\min_{\pi} \; \mathbb{E}_\pi[\mathcal{T}^\star], \tag{5}$$

where the expectation is over training and allocation randomness. The objective accommodates variable epoch costs and is equivalent to minimizing expected epochs to convergence when per-epoch cost is constant and independent of π.

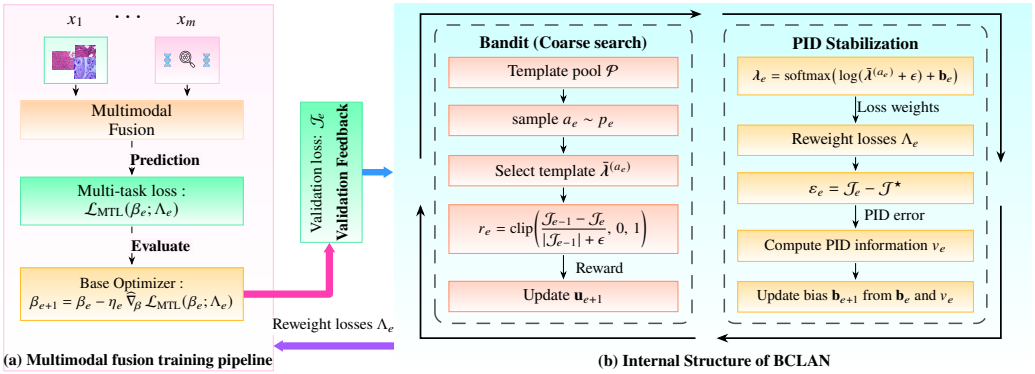

Figure 3: **(a)** Overview of the multimodal fusion optimization pipeline: the fusion backbone is updated by a base optimizer (e.g., `Adam/SGD`) under a simplex-weighted multi-task loss. **(b)** Internal structure of `BCLAN`, a plug-in closed-loop loss-reweighting controller integrated with the fusion backbone: an adversarial bandit selects a coarse loss-weight template and a `PID` module refines it using validation-loss feedback, enabling stable online adaptation without architectural modification. See text for details.

2.3 Bandit-Aware Control Learning

Motivation. Accelerating convergence in multimodal medical fusion motivates *online loss allocation* that couples allocation search with feedback regulation. Bandit-driven mechanisms offer a principled explore–exploit strategy for discovering useful allocations (Liu et al., 2020). In contrast, proportional–integral–derivative (`PID`) control offers complementary *feedback regulation and stabilization* when appropriately tuned (Åström & Murray, 2008), but lacks explicit exploration of alternative allocations. However, such optimization methods primarily address sampling or gradient adaptation, rather than jointly formulating *validation-driven task–modality loss allocation* for multimodal fusion in the medical domain (Liu et al., 2020).

We bridge this gap by introducing **B**andit with **C**ontrol-**L**earning **A**ware optimizatio**N** (`BCLAN`), a lightweight[1] hybrid closed-loop *loss-reweighting module* that augments a standard optimizer (e.g., `Adam/SGD`) without modifying the fusion architecture. To construct adaptive loss allocations, `BCLAN` couples two complementary modules: **(1) Bandit-guided template selection** for online exploration across candidate loss mixtures, and **(2) `PID`-based stabilization** for continuous, bounded refinement in logit space. Their interaction follows an *allocation–training–feedback cycle*: the selected template and available bias form the training allocation, while subsequent validation feedback updates both modules. An overview and pseudo-code are provided in Fig. 3 and App. C.

Simplex-constrained loss allocation. To connect this hybrid design to the learning objective, we collect the task–modality losses into $\mathbf{L}(\beta) = \text{vec}([\mathcal{L}_t^i(\beta)]_{t,i}) \in \mathbb{R}^{Tm}$ and vectorize their weights as $\boldsymbol{\lambda}_e = \text{vec}(\Lambda_e) \in \Delta^{Tm-1}$. Eq. 1 then becomes

$$\mathcal{L}_{\text{MTL}}(\beta; \Lambda_e) = \boldsymbol{\lambda}_e^\top \mathbf{L}(\beta). \tag{6}$$

Within this formulation, the softmax mapping in Eq. 12 preserves nonnegativity and unit sum during online adaptation. When some task–modality losses are inactive, an availability mask $\mathbf{m}_e \in \{0,1\}^{Tm}$ restricts the mixture to active terms:

$$\tilde{\boldsymbol{\lambda}}_e = \mathbf{m}_e \odot \boldsymbol{\lambda}_e, \qquad \boldsymbol{\lambda}_e \leftarrow \frac{\tilde{\boldsymbol{\lambda}}_e}{\|\tilde{\boldsymbol{\lambda}}_e\|_1}. \tag{7}$$

This preserves a valid simplex mixture provided at least one loss component is active.

Training update under adaptive weights. Using this loss mixture, the base optimizer receives the following weighted gradient, with Λ_e held constant during minibatch differentiation:

$$\widehat{\nabla}_\beta \mathcal{L}_{\text{MTL}}(\beta; \Lambda_e) = \sum_{j=1}^{Tm} \lambda_{e,j} \, \widehat{\nabla}_\beta L_j(\beta), \tag{8}$$

where L_j denotes the j-th component of $\mathbf{L}$. The optimizer applies its usual parameter updates, with the allocation controlling the relative influence of task–modality gradients. The remaining design question is how to select these contributions online.

[1] `BCLAN` introduces no additional trainable backbone parameters and updates its controller only at validation intervals; controller overhead is quantified in App. G.

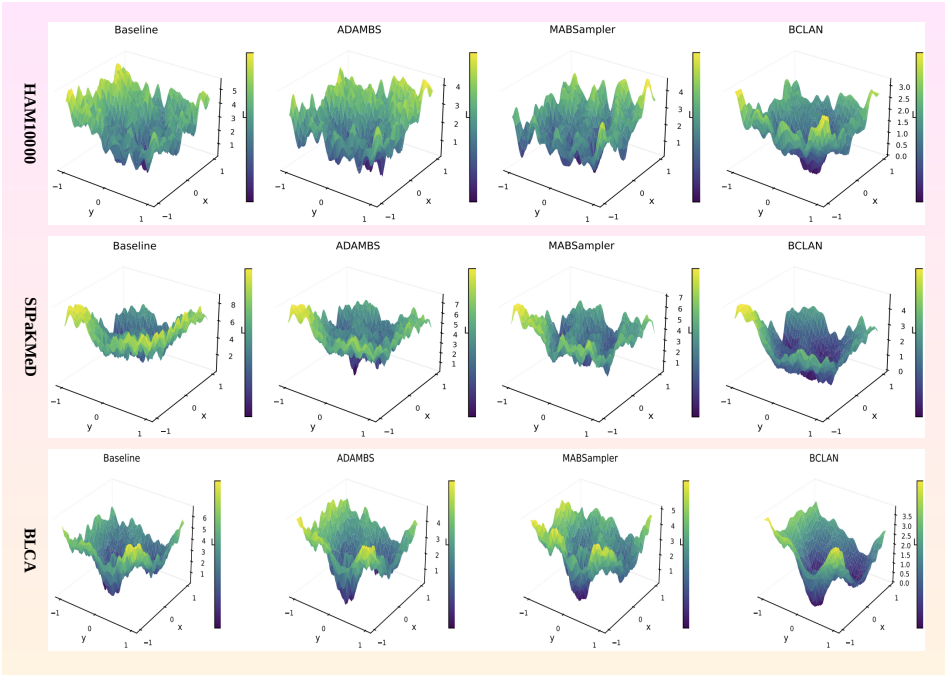

Figure 4: **Loss-landscape comparison on unpaired and paired benchmarks.** 3D visualizations of the (validation) loss surface in a 2D neighborhood of the converged solution for representative unpaired heterogeneous-modality datasets. The horizontal axes (x, y) parameterize displacements along two normalized directions in parameter space, and the vertical axis (L) denotes the resulting loss. Columns compare the baseline optimizer and online adaptive methods (`ADAMBS`, `MABSampler`) against `BCLAN`. Across datasets, `BCLAN` converges to broader, flatter low-loss basins, indicating more stable optimization dynamics under heterogeneous feedback.

1. Bandit-guided template selection. Learning high-dimensional task–modality allocations is challenging under noisy validation feedback (Kocák et al., 2014; Flaxman et al., 2004; Shamir, 2013; Resler & Mansour, 2019). To address this challenge, we propose a *bandit-guided template selection module* that replaces unrestricted exploration of the continuous loss-weight simplex with *online selection among candidate loss mixtures*.

(a) Candidate-template selection. To perform this search, the module maintains a finite pool of loss-weight templates, $\mathcal{P} = \{\bar{\boldsymbol{\lambda}}^{(a)}\}_{a=1}^{A} \subset \Delta^{Tm-1}$. An `EXP3-IX`-style selector (Neu, 2015) samples the template index $a_e \in \{1, \dots, A\}$ through

$$p_e(a) = (1-\gamma)\frac{\mathbf{u}_e(a)}{\|\mathbf{u}_e\|_1} + \frac{\gamma}{A}, \qquad a_e \sim p_e, \tag{9}$$

where $\mathbf{u}_e > 0$ contains the arm weights, initialized as $\mathbf{u}_1 = \mathbf{1}$, and $\gamma \in [0, 1]$ controls uniform exploration. The sampling distribution balances exploration of alternative allocations with exploitation of those associated with validation improvement.

(b) Validation-driven adaptation. To assess learning progress under the hybrid loss allocation, we evaluate the trained model using the fixed-reference validation loss $\mathcal{J}_e$ in Eq. 2. The resulting feedback is translated into a bounded reward measuring relative validation-loss improvement:

$$r_e = \mathrm{clip}\left(\frac{\mathcal{J}_{e-1} - \mathcal{J}_e}{|\mathcal{J}_{e-1}| + \epsilon},\, 0,\, 1\right), \tag{10}$$

where $\mathcal{J}_0$ is evaluated before training and $\epsilon > 0$ prevents division by zero. Normalization expresses progress relative to the preceding loss, while clipping bounds the reward in $[0, 1]$. The resulting reward updates only the selected arm:

$$\mathbf{u}_{e+1}(a) = \begin{cases} \mathbf{u}_e(a) \exp\left(\eta \dfrac{r_e}{p_e(a) + \xi}\right), & a = a_e, \\ \mathbf{u}_e(a), & a \neq a_e, \end{cases} \tag{11}$$

where $\eta > 0$ is the bandit learning rate and $\xi > 0$ regularizes the inverse-probability estimate. A positive reward increases the selected arm's weight, feeding observed learning progress back into subsequent template selection.

Table 1: Performance comparison of online adaptive baselines (e.g., ADAMBS (Liu et al., 2020), MABSampler (Ganapathiraman et al., 2023)) and our hybrid BCLAN across unimodal and multimodal fusion pipelines on *unpaired* heterogeneous-modality medical benchmarks spanning diverse tasks. **Bold**: best.

Datasets →	HAM10000		SIPaKMeD		MORT		CNMC-2019		KVASIR		BRCA	Efficiency Analysis		
Models ↓	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	C-Index	#P	#F	#TTE
NAT	93.1	93.3	91.1	91.5	86.2	88.3	94.1	93.4	89.3	90.1	54.3	20	1.1	22.7
+ ADAMBS	92.9	92.9	91.0	91.2	86.2	88.3	93.9	93.3	89.4	90.1	54.4	20	1.1	15.8
+ MABSampler	93.2	93.4	91.4	91.6	86.3	88.3	94.3	93.5	89.4	90.2	54.5	20	1.1	17.5
+ BCLAN (Ours)	**93.7**	**93.8**	**91.6**	**92.1**	**86.9**	**88.9**	**94.7**	**94.2**	**90.1**	**90.9**	**55.6**	**20**	**1.1**	**12.1**
HEALNet	97.9	98.0	94.7	95.1	91.2	94.3	96.1	96.4	89.7	90.0	57.1	31.6	3.84	32.5
+ ADAMBS	98.0	98.0	94.9	95.4	91.5	94.7	95.9	96.3	89.7	90.1	57.5	31.6	3.84	25.1
+ MABSampler	98.1	98.2	94.9	95.4	91.4	94.6	96.3	96.6	89.9	90.4	57.6	31.6	3.84	29.5
+ BCLAN (Ours)	**98.6**	**98.6**	**95.6**	**96.1**	**92.2**	**95.4**	**96.9**	**97.1**	**90.5**	**90.9**	**58.2**	**31.6**	**3.84**	**18.6**
DRIFA-Net	98.2	98.5	95.6	95.9	91.3	94.1	95.9	96.3	91.9	91.7	56.5	53.8	4.83	27.3
+ ADAMBS	98.4	98.6	95.7	95.9	91.4	94.1	96.1	96.4	91.9	91.7	56.9	53.8	4.83	20.7
+ MABSampler	98.5	98.6	95.6	95.9	91.5	94.3	96.2	96.6	92.1	92.0	57.1	53.8	4.83	22.1
+ BCLAN (Ours)	**99.1**	**99.4**	**96.1**	**96.9**	**92.2**	**94.8**	**96.7**	**97.2**	**92.6**	**92.7**	**59.2**	**53.8**	**4.83**	**15.4**
HyPCA-Net	99.4	99.7	95.7	96.1	92.1	95.2	97.2	97.5	92.4	92.4	60.7	14.47	2.25	30.3
+ ADAMBS	99.4	99.7	95.9	96.1	92.2	95.3	97.4	97.6	92.5	92.5	60.9	14.47	2.25	22.4
+ MABSampler	99.5	99.7	96.0	96.1	92.2	95.3	97.4	97.6	92.6	92.6	60.9	14.47	2.25	24.7
+ BCLAN (Ours)	**99.6**	**99.7**	**96.7**	**96.9**	**93.1**	**95.9**	**97.8**	**98.1**	**93.3**	**93.4**	**61.5**	**14.47**	**2.25**	**18.5**
MMLego	98.1	98.0	94.9	95.2	91.5	94.7	96.1	96.4	89.7	90.0	59.8	102.4	5.36	45.5
+ ADAMBS	98.4	98.4	95.1	95.4	91.7	94.9	96.2	96.5	89.9	90.1	59.9	102.4	5.36	34.2
+ MABSampler	98.5	98.6	95.2	95.2	91.8	94.9	96.2	96.5	90.1	90.1	60.1	102.4	5.36	36.9
+ BCLAN (Ours)	**98.8**	**98.8**	**95.7**	**96.2**	**92.3**	**95.5**	**96.9**	**97.1**	**90.6**	**90.9**	**61.2**	**102.4**	**5.36**	**29.3**
MAIL	99.4	99.7	96.3	96.5	91.4	94.4	97.1	97.6	92.3	92.3	57.1	11.7	1.84	27.1
+ ADAMBS	99.5	99.7	96.3	96.5	91.7	94.8	97.2	97.7	92.4	92.5	57.3	11.7	1.84	20.6
+ MABSampler	99.5	99.7	96.4	96.8	91.6	94.9	97.3	97.8	92.5	92.6	57.4	11.7	1.84	22.4
+ BCLAN (Ours)	**99.6**	**99.9**	**97.1**	**97.2**	**92.5**	**95.6**	**97.9**	**98.5**	**92.8**	**93.2**	**58.4**	**11.7**	**1.84**	**16.8**

2. PID-based stabilization. While the bandit adapts template selection, changes in the resulting loss mixture can introduce weight fluctuations that hinder convergence. To stabilize this online adaptation, we propose complementary PID feedback that regulates an adaptive bias by tracking validation loss relative to $\mathcal{J}^\star$.

(a) Hybrid loss-weight construction. Discrete templates constrain the resolution of candidate allocations. To extend template selection with fine adjustments without enlarging the search pool, we combine the selected template with the available bias $\mathbf{b}_e \in \mathbb{R}^{Tm}$ in logit space:

$$\boldsymbol{\theta}_e = \log\Big(\bar{\boldsymbol{\lambda}}^{(a_e)} + \epsilon\Big) + \mathbf{b}_e, \qquad \boldsymbol{\lambda}_e = \mathrm{softmax}(\boldsymbol{\theta}_e). \tag{12}$$

Here, $\epsilon > 0$ also keeps the logarithm well-defined. After masked renormalization through Eq. 7, these weights define the training mixture in Eq. 6. Thus, the current allocation uses the available bias, while the ensuing validation feedback informs its update.

(b) Validation-loss tracking. To guide refinement using the resulting validation feedback, the controller measures the *tracking deviation*

$$\varepsilon_e = \mathcal{J}_e - \mathcal{J}^\star, \tag{13}$$

where $\mathcal{J}^\star$ is the validation-loss threshold defined in §. 2.2. To incorporate its temporal evolution, the controller combines the instantaneous deviation, its cumulative sum, and its successive change:

$$v_e = K_p\varepsilon_e + K_i\sum_{j=1}^{e}\varepsilon_j + K_d(\varepsilon_e - \varepsilon_{e-1}), \tag{14}$$

with $\varepsilon_0 = \mathcal{J}_0 - \mathcal{J}^\star$. The gains (K_p, K_i, K_d) govern proportional responsiveness, compensation for accumulated deviations, and derivative damping when tuned (Åström & Murray, 2008).

(c) Bounded bias adaptation. To translate this control input into constrained refinement, the controller updates the bias along a directional prior $\mathbf{d} \in \mathbb{R}^{Tm}$ and applies componentwise clipping:

$$\mathbf{b}_{e+1} = \mathrm{clip}(\mathbf{b}_e - \lambda_b v_e \mathbf{d},\, -B,\, B)\,, \tag{15}$$

where $\lambda_b > 0$ is the bias step size, $B > 0$ is the bias bound, and $\mathbf{b}_1 = \mathbf{0}$. Consequently, $\|\mathbf{b}_e\|_\infty \le B$, constraining logit-space departures from the selected template. The updated bias and arm weights jointly determine the next loss allocation, completing the *validation-driven online optimization loop*.

2.4 THEORETICAL ANALYSIS

We connect BCLAN's hybrid loss allocation to *validation-loss convergence* and the associated training-time objective in Eq. 5. Theorem 1 establishes bounded allocation refinement and an expected convergence-time guarantee. Corollary 1 characterizes when the resulting progress advantage exceeds the cost of online adaptation, improving the *training-time efficiency–performance trade-off* at the specified validation-loss criterion.

Theorem 1 (Validation-loss convergence and training-time efficiency)**.** *Consider the loss-based BCLAN controller with $A \geq 2$, $\mathbf{u}_1 = \mathbf{1}$, $\mathbf{b}_1 = \mathbf{0}$, nonempty active sets, and finite, nonnegative validation losses. Let n index controller rounds ending at successive validation epochs. For $0 \leq \mathcal{J}^\star < \mathcal{J}_0$ and $\epsilon > 0$, define*

$$h = \log \frac{\mathcal{J}_0 + \epsilon}{\mathcal{J}^\star + \epsilon}, \qquad \Omega_A = 1 + \frac{(1-\gamma)\log A}{\eta}. \tag{16}$$

Let $0 < \gamma < 1$, $\xi > 0$, and $0 < \eta \leq 2\xi$. Under Assumptions 1–3 in App. B.1, suppose that the cost-normalized net-progress coefficient ρ_{B} in Eq. 28 is positive. Then:

(i) Bounded hybrid refinement. *The allocation remains on the active simplex and $\|\mathbf{b}_n\|_\infty \leq B$. For every active component j,*

$$e^{-2B}\breve{\boldsymbol{\lambda}}_{n,j}^{(a_n)} \leq \lambda_{n,j} \leq e^{2B}\breve{\boldsymbol{\lambda}}_{n,j}^{(a_n)}, \tag{17}$$

where $\breve{\boldsymbol{\lambda}}_n^{(a_n)}$ denotes the selected template after ϵ-smoothing with zero bias.

(ii) Expected convergence time. *The validation-loss criterion is reached in finite expected training time:*

$$\mathbb{E}_{BCLAN}[\mathcal{T}^\star] \leq \frac{h + \Omega_A}{\rho_{\mathrm{B}}}. \tag{18}$$

Here, h measures the required logarithmic loss reduction, Ω_A accounts for finite template search and the final validation round, and ρ_{B} incorporates template progress, validation-loss increases, and elapsed training cost.

Proof. See App. B.2. □

Corollary 1 (Accelerated convergence and the training-time efficiency–performance trade-off)**.** *Consider a static-weight baseline π_0 with the same initialization, backbone, base optimizer, validation criterion, and evaluation schedule. Under Theorem 1 and Assumption 4 in App. B.1, let $\rho_0 > 0$ be the baseline comparison coefficient in Eq. 31. Then*

$$\mathbb{E}_{\pi_0}[\mathcal{T}^\star] - \mathbb{E}_{BCLAN}[\mathcal{T}^\star] \geq h\left(\frac{1}{\rho_0} - \frac{1}{\rho_{\mathrm{B}}}\right) - \frac{\Omega_A}{\rho_{\mathrm{B}}}. \tag{19}$$

In particular, $\rho_{\mathrm{B}} > \rho_0(1 + \Omega_A/h)$ guarantees accelerated convergence in expected training time at the same validation-loss level. The bound separates the benefit of faster cost-normalized progress from the finite cost of learning the allocation, with validation and controller overhead included.

Proof. See App. B.3. □

3 EXPERIMENTAL RESULTS

Datasets. We evaluate BCLAN on 12 heterogeneous medical datasets across four settings: (1) **unpaired imaging**, following MAIL (Dhar et al., 2026c) and DRIFA-Net (Dhar et al., 2025): HAM10000 (Tschandl et al., 2018), SIPaKMeD (Plissiti et al., 2018), CNMC-2019 (Mourya et al., 2019), and KVASIR (Pogorelov et al., 2017); (2) **unpaired omics and EHR**, following CURE (Dhar et al., 2026b): MIMIC-III (MORT) (Johnson et al., 2016) and BRCA (Lingle et al., 2016); (3) **paired WSI–omics**, following HEALNet (Hemker et al., 2024): BLCA (Network et al., 2014), GBMLGG (Bakas et al., 2017), and KIRP (Linehan et al., 2016); and (4) **paired image–EHR and image–text**, following CARZero (Lai et al., 2024): ADNI (Jack Jr et al., 2008), CXR14 (Wang et al., 2017), and Open-I (Demner-Fushman et al., 2016).

Baselines. We compare BCLAN against four online adaptive methods—ADAMBS (Liu et al., 2020), MABSampler (Ganapathiraman et al., 2023), Incremental PID (Wang & Zhang, 2022), and PID-AGD (Zhang & Xie, 2024)—integrated into seven baselines: unimodal NAT (Hassani et al., 2023); vision–language pretraining CARZero (Lai et al., 2024); and multimodal fusion HEALNet (Hemker et al., 2024), DRIFA-Net (Dhar et al., 2025), HyPCA-Net (Dhar et al.,

2026a), MMLego (Hemker et al., 2025), and MAIL (Dhar et al., 2026c). **Bold** denotes best results; following (Hemker et al., 2024), we report mean ± standard deviation over five random seeds.

Evaluation Metrics and Training Details. We report accuracy (ACC), AUC, concordance index (C-Index), number of parameters (#P, in millions), floating-point operations (#F, in GFLOPs), and training time per epoch (#TTE, in seconds). All models are trained for 200 epochs on a single 80GB NVIDIA A100 GPU, following (Dhar et al., 2026c). We use negative log-likelihood (NLL) loss for survival prediction (WSI and multi-omics) following Hemker et al. (2024) and cross-entropy loss for classification (imaging and EHR) following (Dhar et al., 2026c). Optimization uses Adam (initial learning rate 10^{-3}) with ReduceLROnPlateau (minimum learning rate 10^{-6}), following (Dhar et al., 2026c). Additional dataset and implementation details are provided in App. D–E.

3.1 Performance Comparisons

Tables 1–3 compare BCLAN-integrated learners with the corresponding unmodified unimodal learner (e.g., NAT), multimodal fusion pipelines (e.g., CARZero, HEALNet, DRIFA-Net, HyPCA-Net, MMLego, and MAIL), and their available online-adaptation variants (ADAMBS, MABSampler, Incremental PID, and PID-AGD) across twelve medical benchmarks. BCLAN serves as a plug-in closed-loop replacement module for static loss weighting, introducing validation-driven online allocation without modifying the underlying architecture. Across these frameworks, BCLAN improves downstream performance by up to 2.7 points and reduces training time per epoch i.e., #TTE by 30.3%–50.4% relative to the unmodified pipelines. Against the strongest adaptation selected by average predictive performance within each model block, it gains 0.43–0.77 points with further #TTE reductions of 10.3%–36.9% (see Table 13; App. F). These results support its applicability to heterogeneous data in the medical domain for diverse learning tasks.

Table 2: Performance on *paired* heterogeneous-modality medical benchmarks across downstream tasks.

Datasets →	BLCA	GBMLGG	KIRP	Efficiency
Models ↓	C-Index	C-Index	C-Index	#TTE
DRIFA-Net	68.7±7.38	85.5±2.38	80.9±3.43	35.7
+ ADAMBS	68.5±7.38	85.3±2.38	80.1±3.43	24.1
+ MABSampler	68.8±7.38	85.5±2.38	80.9±3.43	26.5
+ Incremental PID	68.6±5.51	85.6±1.63	80.4±2.59	27.5
+ PID-AGD	68.9±6.21	85.7±1.30	80.9±2.64	23.2
+ BCLAN (Ours)	**69.2**±7.38	**86.1**±2.38	**81.5**±3.43	**17.7**
HEALNet	71.4±1.92	85.7±2.27	86.8±5.56	39.2
+ ADAMBS	71.6±1.92	85.9±2.27	86.9±5.56	24.7
+ MABSampler	71.7±1.92	86.1±2.27	87.2±5.56	25.4
+ Incremental PID	71.7±0.87	85.9±2.64	87.2±4.41	32.5
+ PID-AGD	71.7±1.92	86.1±2.27	87.2±5.56	28.8
+ BCLAN (Ours)	**72.1**±1.92	**86.9**±2.27	**87.5**±5.56	**20.3**
MAIL	70.4±1.87	84.2±2.81	87.3±3.52	31.5
+ ADAMBS	70.5±1.93	84.4±3.08	86.7±4.08	24.3
+ MABSampler	70.7±1.28	84.9±2.80	87.4±3.38	26.7
+ Incremental PID	70.6±1.93	84.6±3.08	86.9±4.08	25.2
+ PID-AGD	70.7±1.28	85.1±2.80	87.6±3.38	23.8
+ BCLAN (Ours)	**71.2**±1.19	**85.6**±2.25	**88.1**±2.91	**19.1**

Why the hybrid allocation matters. The gains are consistent with complementary *allocation search and feedback refinement*: the bandit explores loss-weight templates and favors validation-improving allocations, while PID feedback continuously refines their weights through bounded logit-space updates. Together, they adapt task–modality gradient contributions as learning progresses, with fixed-reference validation preserving comparable feedback. Component ablations support this complementarity: joint adaptation improves ACC/AUC by 0.2–0.8 percentage points over either component alone, with 18.0%–26.0% lower reported #TTE (Table 4). Updating only at validation intervals limits controller cost without adding trainable backbone parameters or changing backbone FLOPs, supporting a favorable *performance–training-time efficiency trade-off. More results, ablations and discussions can be found in App. F–H.*

Table 3: Performance on *paired* heterogeneous-modality datasets for CARZero pipeline across downstream tasks.

Datasets →	ADNI	CXR14	Open-I	Efficiency
Models ↓	mAP	AUC	AUC	#TTE
CARZero	84.7 ±0.0068	81.1±0.84	83.8±0.59	251.3
+ ADAMBS	84.7 ±0.0075	81.4±0.54	83.9±0.39	180.7
+ MABSampler	84.9 ±0.0051	81.5±0.42	84.1±0.37	186.2
+ Incremental PID	85.0 ±0.0027	81.2±0.15	84.1±0.64	215.4
+ PID-AGD	85.2 ±0.0025	81.7±0.31	84.2±0.22	195.3
+ BCLAN (Ours)	**85.7** ±0.0018	**82.2**±0.23	**84.6**±0.27	**175.2**

3.2 BCLAN Converges to Broader, Flatter Low-loss Basins

We complement the efficiency–performance comparisons in Tables 1–2 with validation-loss landscapes around each method's converged solution (Fig. 4; App. F.1, Fig. 11). Each 3D surface plots validation loss (L, vertical axis) under weight displacements along two normalized directions (x, y). Broader basins and smoother curvature indicate local robustness—smaller loss changes under small parameter displacements—providing a geometric perspective on optimization stability.

Across unpaired and paired medical datasets, BCLAN consistently reaches broader, flatter low-loss basins than prior online adaptive methods. Bandit-only adaptation instead tends to produce sharper or more rugged geometry, consistent with high-variance feedback reactions when controlling a *loss mixture*. This contrast is consistent with BCLAN's coupling of coarse explore–exploit search and stabilizing PID refinement: damping noisy reweighting favors wider, flatter regions and allows validation-driven loss reallocation without destabilizing convergence across heterogeneous medical modalities. Together, these observations support the controller's well-behaved adaptation and its central principle—*search* for improved allocations while remaining *stable* under noisy feedback—complementing the reported gains in convergence speed and reliability across both regimes.

3.3 Reward-scale Sensitivity and Comparison

We examine sensitivity to the *reward scale* modulating the validation-improvement reward in the EXP3-IX update (see §. 2.3), comparing BCLAN against ADAMBS, MABSampler, Incremental PID, and PID-AGD (Fig. 5). Across unpaired and paired regimes, BCLAN consistently outperforms these baselines throughout the reward-scale sweep (0.1–10), maintaining a clear margin. Multi-run trajectories and shaded 95% confidence bands indicate that these gains persist across random seeds rather than arising from isolated runs. Importantly, BCLAN exhibits only mild sensitivity to reward scaling: smaller scales temper bandit adaptation, whereas larger scales increase reactivity. In both cases, PID refinement stabilizes online loss reweighting, mitigating the instability that can arise when the controlled variable is the *loss mixture* itself. Together, these observations support robustness to reward scaling while preserving the strongest reward-conditioned performance among the compared methods across both multimodal regimes.

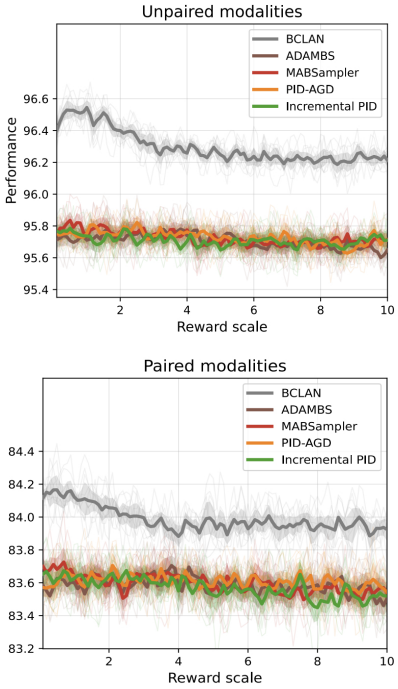

Figure 5: Performance across reward scales for BCLAN vs. baselines on paired and unpaired datasets.

4 Ablation Study

Table 4 isolates BCLAN's two components: the EXP3-IX bandit for template selection and PID for loss-mixture stabilization and refinement. Using a fixed MAIL on unpaired benchmarks, we compare open-loop (neither component), bandit-only, PID-only, and BCLAN. The components provide complementary benefits: bandit-only yields greater training-time savings, whereas PID-only generally yields stronger predictive gains. BCLAN achieves the best ACC/AUC across all tasks and the lowest #TTE, reducing it from 27.1 to 16.8 s while improving MORT AUC from 94.4 to 95.6. These results support coupling efficient coarse exploration with stabilizing PID refinement to translate adaptive loss allocation into improved performance and training efficiency. Further analyses of stability and convergence are provided in App. G.

Table 4: **Module-wise ablations.** Component effects on unpaired benchmarks with a fixed MAIL backbone.

Components		HAM10000		SIPaKMeD		MORT		CNMC-2019		Efficiency
Bandit	PID	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	#TTE
×	×	99.4	99.7	96.3	96.5	91.4	94.4	97.1	97.6	27.1
✓	×	99.4	99.7	96.7	96.7	91.9	94.8	97.5	98.0	20.5
×	✓	99.4	99.7	96.8	96.9	92.1	95.0	97.5	97.9	22.7
✓	✓	**99.6**	**99.9**	**97.1**	**97.2**	**92.5**	**95.6**	**97.9**	**98.5**	**16.8**

5 Conclusion

We presented BCLAN, a hybrid closed-loop framework that couples bandit-based template search with PID-based stabilization for adaptive loss allocation in unimodal and multimodal medical learning. Without altering the architecture, BCLAN consistently improves predictive performance while reducing training time across heterogeneous benchmarks, achieving a favorable training-time efficiency–performance trade-off. A promising future direction is to extend BCLAN to larger multimodal foundation models with dynamically learned template spaces and richer cross-task feedback.

AI USE STATEMENT

We selected few sentences at a time and asked GPT to check the grammar.

REFERENCES

Wangpeng An, Haoqian Wang, Qingyun Sun, Jun Xu, Qionghai Dai, and Lei Zhang. A PID controller approach for stochastic optimization of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8522–8531, 2018.

Yajun An, Jiale Chen, Huan Lin, Zhenbing Liu, Siyang Feng, Hualong Zhang, Rushi Lan, Zaiyi Liu, and Xipeng Pan. CA-MLIF: Cross-attention and multimodal low-rank interaction fusion framework for tumor prognostic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1764–1772, 2025.

Karl Johan Åström and Richard M Murray. Feedback systems: an introduction for scientists and engineers. 2008.

Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific data*, 4(1):1–13, 2017.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27, 2014.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, 2018.

Eduardo Cueto-Mendoza and John D. Kelleher. A framework for measuring the training efficiency of a neural architecture. *Artificial Intelligence Review*, 57(12):349, 2024. doi: 10.1007/s10462-024-10943-8.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016.

Joy Dhar, Nayyar Zaidi, Maryam Haghighat, Sudipta Roy, Puneet Goyal, Azadeh Alavi, and Vikas Kumar. Multimodal fusion learning with dual attention for medical imaging. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4362–4371, 2025. doi: 10.1109/WACV61041.2025.00428.

Joy Dhar, Manish Kumar Pandey, Debashis Das Chakladar, Maryam Haghighat, Azadeh Alavi, Sajib Mistry, and Nayyar Zaidi. Hypca-net: Advancing multimodal fusion in medical image analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1831–1840, 2026a.

Joy Dhar, Manish Kumar Pandey, Nayyar Zaidi, Chen Chen, Maryam Haghighat, Ferdous Sohel, and Puneet Goyal. Advancing multimodal fusion on heterogeneous medical data with hybrid geometry attention. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 863–874, 2026b.

Joy Dhar, Nayyar Zaidi, and Maryam Haghighat. Effective and robust multimodal medical image analysis. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 188–199, 2026c.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

Vignesh Ganapathiraman, Francisco Javier Calderon, and Anila Joshi. Bandit sampling for faster neural network training with sgd. In *International Conference on Learning Representations (ICLR), Tiny Papers Track*, 2023.

Zi Gong, Hang Yu, Cong Liao, Bingchang Liu, Chaoyu Chen, and Jianguo Li. Coba: Convergence balancer for multitask finetuning of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8063–8077, 2024.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6185–6194, 2023.

Konstantin Hemker, Nikola Simidjievski, and Mateja Jamnik. HEALNet: Multimodal fusion for heterogeneous biomedical data. *Advances in Neural Information Processing Systems*, 37:64479–64498, 2024.

Konstantin Hemker, Nikola Simidjievski, and Mateja Jamnik. Multimodal LEGO: Model merging and fusion across topologies and modalities in biomedicine. In *International Conference on Learning Representation ICLR*, 2025.

S. C. Huang, L. Shen, M. P. Lungren, and S. Yeung. Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3942–3951, 2021.

Md Mofijul Islam and Tariq Iqbal. MuMu: Cooperative multitask learning-based guided multimodal fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1043–1051, 2022.

Clifford R Jack Jr, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L. Whitwell, Chadwick Ward, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 27(4):685–691, 2008.

Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. *arXiv preprint arXiv:2103.02631*, 2021.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2525–2534. PMLR, 2018.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tomáš Kocák, Gergely Neu, Michal Valko, and Rémi Munos. Efficient learning by implicit exploration in bandit problems with side observations. *Advances in Neural Information Processing Systems*, 27, 2014.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Haoran Lai, Qingsong Yao, Zihang Jiang, Rongsheng Wang, Zhiyang He, Xiaodong Tao, and S Kevin Zhou. Carzero: Cross-attention alignment for radiology zero-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11137–11146, 2024.

Marston Linehan, Rabindra Gautam, Shanah Kirk, Yueh Lee, Charles Roche, Ermelinda Bonaccio, Joe Filippini, Kimberly Rieger-Christ, John Lemmerman, and Rose Jarosz. The cancer genome atlas cervical kidney renal papillary cell carcinoma collection (tcga-kirp), 2016. URL `https://www.cancerimagingarchive.net/collection/tcga-kirp/`.

Wilma Lingle, Bradley J. Erickson, Margarita L. Zuley, Rose Jarosz, Ermelinda Bonaccio, Joe Filippini, Jose M. Net, Len Levi, Elizabeth A. Morris, Gloria G. Figler, Pierre Elnajjar, Shanah Kirk, Yueh Lee, Maryellen Giger, and Nicholas Gruszauskas. The cancer genome atlas breast invasive carcinoma collection (tcga-brca), 2016. URL `https://www.cancerimagingarchive.net/collection/tcga-brca/`.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

Rui Liu, Tianyi Wu, and Barzan Mozafari. Adam with bandit sampling for deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Simmi Mourya, Sonaal Kant, Pulkit Kumar, Anubha Gupta, and Rita Gupta. All challenge dataset of isbi 2019 (c-nmc 2019), 2019. URL `https://www.cancerimagingarchive.net/collection/c-nmc-2019/`.

Cancer Genome Atlas Research Network et al. Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature*, 507(7492):315, 2014.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

Hossein R Nowdeh, Jie Ji, Xiaolong Ma, and Fatemeh Afghah. Modality-aware sam: Sharpness-aware-minimization driven gradient modulation for harmonized multimodal learning. *arXiv preprint arXiv:2510.24919*, 2025.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8238–8247, 2022.

Maria E Plissiti, Panagiotis Dimitrakopoulos, Giorgos Sfikas, Christophoros Nikou, Orestis Krikoni, and Avraam Charchanti. Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3144–3148. IEEE, October 2018.

Roman Pogodin and Tor Lattimore. On first-order bounds, variance and gap-dependent bounds for adversarial bandits. In *Uncertainty in Artificial Intelligence*, pp. 894–904. PMLR, 2020.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. KVASIR: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of MMSys '17*, pp. 1–6, 2017. doi: 10.1145/3083187.3083212.

Alon Resler and Yishay Mansour. Adversarial online learning with noise. In *International Conference on Machine Learning*, pp. 5429–5437. PMLR, 2019.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In Shai Shalev-Shwartz and Ingo Steinwart (eds.), *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pp. 3–24, Princeton, NJ, USA, 12–14 Jun 2013. PMLR. URL https://proceedings.mlr.press/v30/Shamir13.html.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *International conference on machine learning*, pp. 9133–9143. PMLR, 2020.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1): 1–9, 2018.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12692–12702, 2020.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Zenghui Wang and Jun Zhang. Incremental pid controller-based learning rate scheduler for Stochastic Gradient Descent. *IEEE transactions on neural networks and learning systems*, 35(5):7060–7071, 2022.

Zenghui Wang and Jun Zhang. Incremental pid controller-based learning rate scheduler for stochastic gradient descent. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):7060–7071, 2024.

Shicai Wei, Chunbo Luo, and Yang Luo. Improving multimodal learning via imbalanced learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2250–2259, 2025.

Yang Yang, Fengqiang Wan, Qing-Yuan Jiang, and Yi Xu. Facilitating multimodal classification via dynamically learning modality gap. *Advances in Neural Information Processing Systems*, 37: 62108–62122, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Siyuan Zhang and Linbo Xie. A pid controller approach for adaptive probability-dependent gradient decay in model calibration. *Advances in Neural Information Processing Systems*, 37:114744–114764, 2024.

Xiaohui Zhang, Jaehong Yoon, Mohit Bansal, and Huaxiu Yao. Multimodal representation learning by alternating unimodal adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 27456–27466, 2024.

APPENDIX

A RELATED STUDY

Multimodal Fusion Architectures. Multimodal fusion models such as GLORIA (Huang et al., 2021) learn shared representations from heterogeneous medical sources, but limited cross-modal interactions can constrain their effectiveness (Dhar et al., 2025; 2026a;c). Recent approaches, including DRIFA-Net (Dhar et al., 2025), HEALNet (Hemker et al., 2024), MMLego (Hemker et al., 2025), and CA-MLIF (An et al., 2025), pursue richer coarse- and fine-grained features through more expressive cross-modal interactions. Although attention-driven refinement strengthens representation learning, cascaded designs may attenuate information and increase computational cost (Dhar et al., 2026a;c). HyPCA (Dhar et al., 2026a) and MAIL (Dhar et al., 2026c) further improve robustness and architectural efficiency through dual-domain or parallel interaction learning. As reflected in the architecture-focused entries of Table 5, these advances primarily concern *representation and fusion*, rather than online optimization. Architectural efficiency therefore leaves a complementary challenge: adapting task–modality loss contributions to accelerate validation convergence, particularly under constrained training budgets.

Online Adaptive Optimization. A complementary line of work adapts the training process through curriculum and self-paced learning (Bengio et al., 2009; Kumar et al., 2010), adaptive sampling (Chang et al., 2017; Katharopoulos & Fleuret, 2018), and loss- or gradient-balancing mechanisms (Kendall et al., 2018; Chen et al., 2018; Yu et al., 2020; Liu et al., 2023; Gong et al., 2024). Bandit-based methods, including ADAMBS and MABSampler, introduce explicit explore–exploit strategies for adapting sampling or update policies, with demonstrated benefits in unimodal learning (Liu et al., 2020; Ganapathiraman et al., 2023) (Table 5). However, transferring this principle from sampling to a *task–modality loss mixture* raises a distinct stability challenge: noisy, non-stationary validation feedback can induce high-variance updates and abrupt loss reallocations (Kocák et al., 2014; Pogodin & Lattimore, 2020; Besbes et al., 2014; Resler & Mansour, 2019).

Table 5: Research-gap comparison of representative approaches. "Online learning" denotes feedback-driven adaptation during training; "time-aware efficiency" concerns training convergence or elapsed training cost rather than architectural complexity alone. Checkmarks indicate coverage in this comparison; blank entries do not imply inapplicability.

Approach	**Online** Learning	**Time-Aware** Efficiency	**Unimodal** Analysis	**Multimodal** Analysis
Perceiver				✓
GLORIA				✓
HAMLET				✓
MuMu				✓
M^3Att				✓
MOTCAT				✓
DRIFA-Net				✓
HEALNet				✓
ADAMBS	✓	✓	✓	
MABSampler	✓	✓	✓	
Incremental-PID	✓	✓	✓	
PID-AGD	✓	✓	✓	
BCLAN (Ours)	✓	✓	✓	✓

Feedback-control methods, including the Incremental-PID and PID-AGD approaches in Table 5, offer complementary regulation through tracking-error correction and damping (Åström & Murray, 2008; An et al., 2018; Wang & Zhang, 2024; Zhang & Xie, 2024; Stooke et al., 2020). Yet these controllers do not explicitly explore alternative allocations. Thus, exploration and feedback regulation address complementary requirements, while either mechanism alone may struggle to sustain fast, reliable convergence as task–modality dynamics evolve.

Research Gap. Table 5 highlights the separation between multimodal representation learning and online training adaptation: the listed fusion architectures emphasize multimodal analysis, whereas the selected bandit- and PID-based studies emphasize online efficiency in unimodal settings. The gap addressed here is their integration through a unified closed-loop framework that can *search* for improved loss allocations, *stabilize* their adaptation under heterogeneous feedback, and explicitly address training-time efficiency across paired and unpaired medical data.

BCLAN couples bandit exploration over loss-weight templates with bounded PID refinement in logit space, using validation feedback without modifying the fusion backbone. Within the comparison in Table 5, it jointly covers online learning, time-aware efficiency, unimodal analysis, and multimodal analysis, complementing architectural advances with adaptive loss allocation to improve the training-time efficiency–performance trade-off.

Table 6: Dataset description.

Regime	Dataset	Modalities	BCLAN endpoint / metric	Public scale (native unit)	Target space / #classes
Unpaired imaging	HAM10000	Dermoscopic images	Classification (ACC/AUC)	10,015 images	7 lesion classes
Unpaired imaging	SIPaKMeD	Cervical cytology images	Classification (ACC/AUC)	4,049 isolated-cell images cropped from 966 cluster images	5 classes
Unpaired imaging	C-NMC 2019	Blood-smear single-cell microscopy	Classification (ACC/AUC)	10,661 labeled training cells + 1,867 preliminary test cells	2 classes (normal vs. leukemic B-lymphoblast)
Unpaired imaging	Kvasir	GI endoscopy images	Classification (ACC/AUC)	4,000 labeled images in the original cited release	8 classes
Unpaired EHR	MIMIC-III (MORT)	ICU EHR	Mortality prediction (ACC/AUC)	38,597 adult patients, 49,785 hospital admissions, and 53,423 ICU stays	Binary mortality
Unpaired omics	TCGA-BRCA	Omics	Survival prediction (C-Index)	1,098 cases	Time-to-event + censoring
Paired WSI + omics	TCGA-BLCA	Pathology slides + omics	Survival prediction (C-Index)	412 cases	Time-to-event + censoring
Paired WSI + omics	GBMLGG	Pathology slides + omics	Survival prediction (C-Index)	617 TCGA-GBM cases + 516 TCGA-LGG cases (1,133 raw project cases)	Time-to-event + censoring
Paired WSI + omics	TCGA-KIRP	Pathology slides + omics	Survival prediction (C-Index)	291 cases	Time-to-event + censoring
Paired image + clinical	ADNI	Neuroimaging + EHR	Classification (mAP)	Longitudinal program; ADNI-1 planned ∼800 participants	Paper-specific diagnostic label map
Paired chest imaging	CXR14	Chest X-ray + EHR	Classification (AUC)	112,120 images from 30,805 patients	14 disease labels + No Finding
Paired image + report	Open-I	Chest X-ray + report	Classification (AUC)	8,121 images and 3,996 de-identified reports	Derived label space not fixed in the source corpus

Table 7: BCLAN controller settings and feedback definition.

Setting	Value
Reference validation weighting Λ_{ref}	Fixed uniform reference weighting
Initial template distribution $q_1(a)$	Uniform over $\mathcal{P}$
Template pool $\mathcal{P}$	$\{[0.50, 0.50], [0.65, 0.35], [0.35, 0.65], [0.80, 0.20], [0.20, 0.80]\}$
Bandit exploration rate γ	0.07
Bandit learning rate η	$0.5 \quad (0.5/K,\, K = 5)$
EXP3-IX bias ξ	0.05
Controller feedback	Mean validation accuracy
Controller target $s^\star$	0.88
K_p	2.0
K_i	0.4
K_d	0.2
Bias step size λ_b	0.20
Bias clipping B	± 2.0
Integral clipping	± 50.0
Validation interval Δ_v	2

B Additional Theoretical Analysis

The analysis connects BCLAN's allocation mechanism to its training-time objective through three complementary results: bounded template refinement, cumulative reward transfer, and validation-loss progress. Together, these results establish the expected convergence-time bound in Theorem 1. A subsequent comparison with static weighting yields Corollary 1, identifying when the convergence benefit exceeds the cost of online adaptation.

B.1 Assumptions and Lemmas

The analysis fixes the initial model and optimizer state, training and validation data, uniform reference weighting Λ^{ref}, and deterministic validation schedule $0 = e_0 < e_1 < e_2 < \cdots$. Controller round n comprises template selection, training under the resulting hybrid allocation, and validation at epoch e_n. Controller quantities use the round index n, while $\mathcal{J}_{e_n}$ retains its training-epoch index.

Let $\mathcal{H}_{n-1}$ denote the training and controller information available before template selection in round n. The sampling distribution p_n, current bias $\mathbf{b}_n$, and prescribed active set $\mathcal{I}_n$ are determined from

this information. Conditional expectations given $\mathcal{H}_{n-1}$ average over subsequent template-selection and training randomness. All expectations also condition on the fixed initialization and data.

For finite, nonnegative validation losses and $0 \le \mathcal{J}^\star < \mathcal{J}_0$, define the auxiliary convergence round

$$\sigma = \inf\{n \ge 1 : \mathcal{J}_{e_n} \le \mathcal{J}^\star\}, \tag{20}$$

with $\inf \varnothing = \infty$. This is also the first round at which the best-so-far validation loss satisfies the criterion. Whether round n is included depends only on validation evaluations completed before that round. Thus, $\{\sigma \ge n\} \in \mathcal{H}_{n-1}$, and its indicator is determined before template selection.

To retain the computational cost of every training epoch, let

$$C_n = \sum_{e=e_{n-1}+1}^{e_n} \tau_e \tag{21}$$

be the positive, finite elapsed cost of round n, including training, validation, and controller computations. On $\{\sigma < \infty\}$, $\mathcal{T}^\star = \sum_{n=1}^{\sigma} C_n$; otherwise, $\mathcal{T}^\star = \infty$.

The loss-improvement reward in Eq. 10, evaluated between successive validation epochs, becomes

$$r_n = \frac{[\mathcal{J}_{e_{n-1}} - \mathcal{J}_{e_n}]_+}{\mathcal{J}_{e_{n-1}} + \epsilon} \in [0, 1), \tag{22}$$

where $[x]_+ = \max\{x, 0\}$ and $\epsilon > 0$ is the smoothing constant used by the controller. Define

$$\mu_n(a) = \mathbb{E}[r_n \mid \mathcal{H}_{n-1}, a_n = a], \qquad w_n = \left[\log \frac{\mathcal{J}_{e_n} + \epsilon}{\mathcal{J}_{e_{n-1}} + \epsilon}\right]_+. \tag{23}$$

The conditional reward $\mu_n(a)$ includes the available `PID` bias and training randomness. The analytical quantity w_n records upward logarithmic loss variation.

These quantities distinguish the progress available from template selection from the validation-loss increases that can offset it. The following assumptions connect this distinction to expected convergence time.

Assumption 1 (Template-progress coverage)**.** *There exist $\mu > 0$ and a template index $a^\dagger \in \{1, \dots, A\}$, fixed independently of the random training trajectory, such that*

$$\mu_n(a^\dagger) \ge \mu \qquad \textit{almost surely on } \{\sigma \ge n\}, \quad n \ge 1. \tag{24}$$

The candidate need not maximize reward; its conditional expected progress must remain positive at each round begun before `BCLAN` satisfies the convergence criterion, including the round in which the criterion is reached. Because the reward measures positive loss improvements, we separately account for upward validation-loss variation.

Assumption 2 (Controlled upward loss variation)**.** *There exists $\nu \ge 0$ such that*

$$\mathbb{E}[w_n \mid \mathcal{H}_{n-1}] \le \nu \qquad \textit{almost surely on } \{\sigma \ge n\}, \quad n \ge 1. \tag{25}$$

This condition permits nonmonotone validation trajectories; a nonincreasing trajectory admits $\nu = 0$. To express progress in elapsed training time, we also account for the computational cost of each controller round.

Assumption 3 (Finite conditional training cost)**.** *There exists $0 < c_{\mathrm{B}} < \infty$ such that*

$$\mathbb{E}[C_n \mid \mathcal{H}_{n-1}] \le c_{\mathrm{B}} \qquad \textit{almost surely on } \{\sigma \ge n\}, \quad n \ge 1. \tag{26}$$

For $0 < \gamma < 1$, $\xi > 0$, and $0 < \eta \le 2\xi$, define the reward-transfer factor

$$\alpha_A = \frac{(1-\gamma)\gamma}{\gamma + A\xi} \tag{27}$$

and the cost-normalized net-progress coefficient

$$\rho_{\mathrm{B}} = \frac{\alpha_A \mu - \nu}{c_{\mathrm{B}}}. \tag{28}$$

The convergence theorem requires $\alpha_A \mu > \nu$. These analytical coefficients are specified for the validation-loss criterion $\mathcal{J}^\star$; ρ_{B} combines the transferred template progress, upward loss variation, and expected round cost.

The preceding assumptions characterize BCLAN's expected convergence time. To establish acceleration relative to static weighting, we additionally characterize the baseline's progress and cost along its own training trajectory.

Assumption 4 (Static-weight comparator)**.** *The baseline π_0 uses a static loss allocation and has the same initialization, backbone, base optimizer, reference validation criterion, and evaluation schedule as BCLAN. Its validation losses are finite and nonnegative, and its expected convergence time is finite.*

Let $\mathcal{H}_{n-1}^{(0)}$ denote the baseline's training information available before round n, and let $C_n^{(0)}$ denote its round cost. Define

$$\sigma_0 = \inf\left\{ n \geq 1 : \mathcal{J}_{e_n}^{(0)} \leq \mathcal{J}^\star \right\}.$$

There exist $\kappa_0 > 0$ and $c_0 > 0$ such that

$$\mathbb{E}_{\pi_0}\left[\left[\log \frac{\mathcal{J}_{e_{n-1}}^{(0)} + \epsilon}{\mathcal{J}_{e_n}^{(0)} + \epsilon} \right]_+ \,\middle|\, \mathcal{H}_{n-1}^{(0)} \right] \leq \kappa_0, \tag{29}$$

$$\mathbb{E}_{\pi_0}\left[C_n^{(0)} \mid \mathcal{H}_{n-1}^{(0)} \right] \geq c_0, \tag{30}$$

almost surely on $\{\sigma_0 \geq n\}$ for every $n \geq 1$.

The corresponding baseline comparison coefficient is

$$\rho_0 = \frac{\kappa_0}{c_0}, \tag{31}$$

combining an upper bound on positive logarithmic progress with a lower bound on expected round cost.

We first establish the constraint imposed by the hybrid allocation rule. The following lemma shows that normalization preserves simplex feasibility, while bias clipping bounds the multiplicative departure from the selected template.

Lemma 1 (Feasibility and bounded template refinement)**.** *Let $\mathcal{I}_n$ be the nonempty active set in round n. With $\epsilon > 0$, $\mathbf{b}_1 = \mathbf{0}$, and componentwise bias clipping to $[-B, B]$, the allocation remains on the active simplex and $\|\mathbf{b}_n\|_\infty \leq B$. Define the zero-bias projected template*

$$\breve{\lambda}_{n,j}^{(a)} = \begin{cases} \dfrac{\bar{\lambda}_j^{(a)} + \epsilon}{\sum_{k \in \mathcal{I}_n} (\bar{\lambda}_k^{(a)} + \epsilon)}, & j \in \mathcal{I}_n, \\ 0, & j \notin \mathcal{I}_n. \end{cases} \tag{32}$$

Then every active component satisfies

$$e^{-2B} \breve{\lambda}_{n,j}^{(a_n)} \leq \lambda_{n,j} \leq e^{2B} \breve{\lambda}_{n,j}^{(a_n)}.$$

Proof. The initialization and clipped update imply $\|\mathbf{b}_n\|_\infty \leq B$ by induction. For an active component, the hybrid construction gives

$$\begin{aligned} \lambda_{n,j} &= \frac{(\bar{\lambda}_j^{(a_n)} + \epsilon) e^{b_{n,j}}}{\sum_{k \in \mathcal{I}_n} (\bar{\lambda}_k^{(a_n)} + \epsilon) e^{b_{n,k}}} \\ &= \breve{\lambda}_{n,j}^{(a_n)} \frac{e^{b_{n,j}}}{\sum_{k \in \mathcal{I}_n} \breve{\lambda}_{n,k}^{(a_n)} e^{b_{n,k}}}. \end{aligned}$$

Both $e^{b_{n,j}}$ and its convex average in the denominator belong to $[e^{-B}, e^{B}]$. Their ratio therefore lies in $[e^{-2B}, e^{2B}]$, proving the componentwise bound.

Positive smoothing and a nonempty active set ensure a strictly positive normalizing denominator. The active weights are nonnegative and sum to one, while inactive weights are zero. □

Having characterized the admissible refinement, we next analyze how template selection converts available progress into accumulated reward. The following lemma relates the observed reward to the conditional reward of a fixed candidate over bounded stopping horizons, enabling its application to a convergence-dependent training duration.

Lemma 2 (Multiplicative reward transfer)**.** *Consider the positive-exponential reward update in Eq. 11, with* $\mathbf{u}_1 = \mathbf{1}$*,* $0 < \gamma < 1$*,* $\xi > 0$*, and* $0 < \eta \le 2\xi$*. Let* S *be a nonnegative, integer-valued stopping time bounded by a deterministic integer, with* $\{n \le S\} \in \mathcal{H}_{n-1}$*. For every fixed template index* a*,*

$$\mathbb{E}\left[\sum_{n=1}^{S} r_n\right] \ge \alpha_A \mathbb{E}\left[\sum_{n=1}^{S} \mu_n(a)\right] - \frac{(1-\gamma)\log A}{\eta}. \tag{33}$$

Proof. Let

$$W_n = \sum_{a=1}^{A} u_n(a), \qquad q_n(a) = \frac{u_n(a)}{W_n}, \qquad p_n(a) = (1-\gamma)q_n(a) + \frac{\gamma}{A},$$

and define

$$\widehat{r}_n(a) = \frac{r_n \mathbf{1}\{a_n = a\}}{p_n(a) + \xi}.$$

The update is $u_{n+1}(a) = u_n(a)\exp(\eta \widehat{r}_n(a))$. A common normalization of all arm weights leaves the sampling distribution unchanged, so equivalent unnormalized weights may be used in the proof.

For $0 \le x < 2$,

$$e^x - 1 \le \frac{x}{1 - x/2}. \tag{34}$$

Indeed, the function

$$f(x) = \log \frac{1 + x/2}{1 - x/2} - x$$

satisfies $f(0) = 0$ and $f'(x) = x^2/(4 - x^2) \ge 0$. Hence, $e^x \le (1 + x/2)/(1 - x/2)$, yielding Eq. 34.

Only the selected arm changes in round n. Writing

$$x_n = \frac{\eta r_n}{p_n(a_n) + \xi},$$

the conditions $r_n \le 1$, $p_n(a_n) > 0$, and $\eta \le 2\xi$ imply $0 \le x_n < 2$. Therefore,

$$\begin{aligned}
\log \frac{W_{n+1}}{W_n} &= \log(1 + q_n(a_n)(e^{x_n} - 1)) \\
&\le q_n(a_n) \frac{x_n}{1 - x_n/2} \\
&= \frac{\eta q_n(a_n) r_n}{p_n(a_n) + \xi - \eta r_n/2} \\
&\le \frac{\eta q_n(a_n) r_n}{p_n(a_n)} \le \frac{\eta r_n}{1 - \gamma}.
\end{aligned} \tag{35}$$

The penultimate inequality uses $\xi - \eta r_n/2 \ge 0$; the last follows from $p_n(a) \ge (1-\gamma)q_n(a)$.

Uniform initialization gives $W_1 = A$. For any fixed index a and every realized value of S,

$$\log W_{S+1} \ge \log u_{S+1}(a) = \eta \sum_{n=1}^{S} \widehat{r}_n(a).$$

Combining this lower bound with the sum of Eq. 35 yields

$$\sum_{n=1}^{S} r_n \ge (1-\gamma) \sum_{n=1}^{S} \widehat{r}_n(a) - \frac{(1-\gamma)\log A}{\eta}. \tag{36}$$

To relate the estimator to template progress, condition on $\mathcal{H}_{n-1}$:

$$\begin{aligned}\mathbb{E}[\widehat{r}_n(a) \mid \mathcal{H}_{n-1}] &= \frac{p_n(a)}{p_n(a)+\xi}\mu_n(a) \\ &\geq \frac{\gamma}{\gamma + A\xi}\mu_n(a),\end{aligned}$$

using $p_n(a) \geq \gamma/A$ and the monotonicity of $p/(p+\xi)$.

Let M be a deterministic upper bound on S. The indicator $\mathbf{1}\{n \leq S\}$ is determined by $\mathcal{H}_{n-1}$ and can therefore be taken outside the corresponding conditional expectation:

$$\begin{aligned}\mathbb{E}\left[\sum_{n=1}^{S} \widehat{r}_n(a)\right] &= \sum_{n=1}^{M} \mathbb{E}[\mathbf{1}\{n \leq S\}\mathbb{E}[\widehat{r}_n(a) \mid \mathcal{H}_{n-1}]] \\ &\geq \frac{\gamma}{\gamma + A\xi}\mathbb{E}\left[\sum_{n=1}^{S} \mu_n(a)\right].\end{aligned}$$

Taking expectations in Eq. 36 and substituting α_A proves the result. □

The reward-transfer bound characterizes the feedback retained by online selection. Connecting this feedback to convergence requires accounting for validation-loss increases, which the clipped reward does not penalize. The following lemma incorporates these increases through w_n, linking cumulative reward to net logarithmic loss reduction.

Lemma 3 (Validation progress under nonmonotone losses)**.** *For the reward in Eq. 22,*

$$\log \frac{\mathcal{J}_0 + \epsilon}{\mathcal{J}_{e_N} + \epsilon} = \sum_{n=1}^{N} \{-\log(1 - r_n) - w_n\} \geq \sum_{n=1}^{N} (r_n - w_n). \tag{37}$$

For $S_N = \min\{\sigma, N\}$,

$$\sum_{n=1}^{S_N} r_n \leq h + 1 + \sum_{n=1}^{S_N} w_n, \tag{38}$$

where h is defined in Eq. 16.

Proof. When the validation loss decreases or remains unchanged, $w_n = 0$ and

$$1 - r_n = \frac{\mathcal{J}_{e_n} + \epsilon}{\mathcal{J}_{e_{n-1}} + \epsilon}.$$

When the loss increases, $r_n = 0$ and

$$\log \frac{\mathcal{J}_{e_{n-1}} + \epsilon}{\mathcal{J}_{e_n} + \epsilon} = -w_n.$$

Thus, in both cases,

$$\log \frac{\mathcal{J}_{e_{n-1}} + \epsilon}{\mathcal{J}_{e_n} + \epsilon} = -\log(1 - r_n) - w_n.$$

Summation telescopes, and $-\log(1 - r_n) \geq r_n$ proves Eq. 37.

For every prefix $m < \sigma$, $\mathcal{J}_{e_m} > \mathcal{J}^\star$, giving

$$\sum_{n=1}^{m} r_n \leq \log \frac{\mathcal{J}_0 + \epsilon}{\mathcal{J}_{e_m} + \epsilon} + \sum_{n=1}^{m} w_n < h + \sum_{n=1}^{m} w_n.$$

If the criterion is satisfied by round N, the final round contributes at most one additional unit of reward. Otherwise, this inequality applies at $m = N$. Both cases, including the empty prefix when $\sigma = 1$, give Eq. 38. □

B.2 Proof of Theorem 1

The preceding lemmas provide complementary bounds on accumulated reward: template selection supplies a progress lower bound, while the validation criterion limits the reward accumulated before convergence. Combining them controls the expected training duration.

Proof. Part (i) follows from Lemma 1. For part (ii), let $S_N = \min\{\sigma, N\}$. Since $\{n \le S_N\}$ is determined before the template is sampled, the stopped reward-transfer lemma applies.

Using the index $a^\dagger$ from Assumption 1, Lemma 2 gives

$$\alpha_A \mu \, \mathbb{E}[S_N] \le \mathbb{E}\left[\sum_{n=1}^{S_N} r_n\right] + \frac{(1-\gamma)\log A}{\eta}. \tag{39}$$

Meanwhile, Assumption 2 implies

$$\begin{aligned}\mathbb{E}\left[\sum_{n=1}^{S_N} w_n\right] &= \sum_{n=1}^{N} \mathbb{E}[\mathbf{1}\{n \le S_N\}\mathbb{E}[w_n \mid \mathcal{H}_{n-1}]] \\ &\le \nu \, \mathbb{E}[S_N].\end{aligned}$$

Combining these estimates with Eq. 38 separates transferred template progress from upward loss variation:

$$(\alpha_A \mu - \nu)\mathbb{E}[S_N] \le h + 1 + \frac{(1-\gamma)\log A}{\eta} = h + \Omega_A. \tag{40}$$

Because $\alpha_A \mu - \nu > 0$, this bound is uniform in N. As $S_N \uparrow \sigma$, monotone convergence yields

$$\mathbb{E}[\sigma] \le \frac{h + \Omega_A}{\alpha_A \mu - \nu} < \infty.$$

Hence, the validation-loss criterion is satisfied almost surely.

The preceding estimate controls the expected number of controller rounds. To recover the training-time objective, we accumulate the cost of every included round. Because $\{\sigma \ge n\} \in \mathcal{H}_{n-1}$, its indicator can be taken outside the conditional expectation:

$$\begin{aligned}\mathbb{E}_{\texttt{BCLAN}}[\mathcal{T}^\star] &= \sum_{n \ge 1} \mathbb{E}[\mathbf{1}\{\sigma \ge n\} C_n] \\ &= \sum_{n \ge 1} \mathbb{E}[\mathbf{1}\{\sigma \ge n\} \mathbb{E}[C_n \mid \mathcal{H}_{n-1}]] \\ &\le c_{\mathrm{B}} \sum_{n \ge 1} \Pr(\sigma \ge n) \\ &= c_{\mathrm{B}} \mathbb{E}[\sigma] \\ &\le \frac{c_{\mathrm{B}}(h + \Omega_A)}{\alpha_A \mu - \nu} = \frac{h + \Omega_A}{\rho_{\mathrm{B}}}.\end{aligned}$$

The interchange of expectation and summation follows from nonnegative costs; the inequality uses Assumption 3. This proves Eq. 18. □

B.3 Proof of Corollary 1

The theorem bounds the time required by BCLAN to satisfy the validation-loss criterion. To establish acceleration, we derive a lower bound for static weighting at the same criterion and compare the two training-time bounds.

Proof. For the baseline, $\{\sigma_0 \geq n\}$ is determined by the training information available before round n. Applying Assumption 4 therefore gives

$$\begin{aligned}\mathbb{E}_{\pi_0}[\mathcal{T}^\star] &= \sum_{n\geq 1} \mathbb{E}_{\pi_0}\left[\mathbf{1}\{\sigma_0 \geq n\} C_n^{(0)}\right] \\ &\geq c_0 \sum_{n\geq 1} \Pr_{\pi_0}(\sigma_0 \geq n) \\ &= c_0 \mathbb{E}_{\pi_0}[\sigma_0].\end{aligned}$$

The assumed finite expected baseline training time therefore implies $\mathbb{E}_{\pi_0}[\sigma_0] < \infty$.

To bound this duration from below, we compare the required logarithmic loss reduction with the positive progress accumulated along the baseline's trajectory. At the round satisfying the criterion,

$$\begin{aligned}h &\leq \log \frac{\mathcal{J}_0 + \epsilon}{\mathcal{J}_{e_{\sigma_0}}^{(0)} + \epsilon} \\ &= \sum_{n=1}^{\sigma_0} \log \frac{\mathcal{J}_{e_{n-1}}^{(0)} + \epsilon}{\mathcal{J}_{e_n}^{(0)} + \epsilon} \\ &\leq \sum_{n=1}^{\sigma_0} \left[\log \frac{\mathcal{J}_{e_{n-1}}^{(0)} + \epsilon}{\mathcal{J}_{e_n}^{(0)} + \epsilon}\right]_+ .\end{aligned}$$

The inclusion indicator is determined by $\mathcal{H}_{n-1}^{(0)}$. Taking conditional expectations and applying Eq. 29 over the included rounds yields

$$h \leq \kappa_0 \mathbb{E}_{\pi_0}[\sigma_0].$$

The nonnegative summands justify passing from finite truncations to the complete stopped sum. Consequently,

$$\mathbb{E}_{\pi_0}[\mathcal{T}^\star] \geq \frac{c_0 h}{\kappa_0} = \frac{h}{\rho_0}. \tag{41}$$

Both bounds now refer to the same validation-loss criterion and include their respective computational costs. Subtracting BCLAN's upper bound isolates the progress benefit and the finite allocation-search penalty:

$$\begin{aligned}\mathbb{E}_{\pi_0}[\mathcal{T}^\star] - \mathbb{E}_{\texttt{BCLAN}}[\mathcal{T}^\star] &\geq \frac{h}{\rho_0} - \frac{h + \Omega_A}{\rho_{\mathrm{B}}} \\ &= h\left(\frac{1}{\rho_0} - \frac{1}{\rho_{\mathrm{B}}}\right) - \frac{\Omega_A}{\rho_{\mathrm{B}}}.\end{aligned}$$

This lower bound is strictly positive when

$$\rho_{\mathrm{B}} > \rho_0 \left(1 + \frac{\Omega_A}{h}\right).$$

Under this condition, the cost-normalized progress advantage exceeds the finite search penalty, establishing a strict reduction in expected elapsed training time. □

B.4 Training-Time Interpretation

The theorem and corollary characterize the training-time efficiency–performance trade-off at the fixed-reference validation-loss level $\mathcal{J}^\star$. Under Theorem 1, an expected training-time budget $\mathcal{B} > 0$ is sufficient whenever

$$\rho_{\mathrm{B}} \mathcal{B} \geq h + \Omega_A \quad \Longrightarrow \quad \mathbb{E}_{\texttt{BCLAN}}[\mathcal{T}^\star] \leq \mathcal{B}.$$

Corollary 1 further identifies when adaptive allocation improves this cost relative to static weighting, accounting for both the progress advantage and the computational cost of online adaptation.

Table 8: Pairwise dominance of BCLAN over competing adaptation families, computed from Tables 1–3. "Dataset score" averages all metrics reported for a given dataset–backbone pair; "Backbone overall" averages all performance metrics reported for a backbone block.

Competing method	Metric-cell wins/ties ↑	Dataset-score wins/ties ↑	Backbone-overall wins/ties ↑	Faster #TTE ↑
Baseline	77+1/78	48/48	10/10	10/10
ADAMBS	77+1/78	48/48	10/10	10/10
MABSampler	77+1/78	48/48	10/10	10/10
Incremental PID	12/12	12/12	4/4	4/4
PID-AGD	12/12	12/12	4/4	4/4

C Pseudo-code Analysis

Alg. 1 instantiates BCLAN as an outer-loop controller that reallocates optimization effort across task–modality losses without modifying the fusion backbone. It combines coarse bandit search with continuous PID refinement through an *allocation–training–feedback cycle*. Controller round n spans epochs $e_{n-1}+1,\dots,e_n$ and ends with validation; the arm weights and refinement bias are updated once per validation interval.

Bandit-guided template search. At the start of round n, an EXP3−IX-style selector constructs the sampling distribution p_n over the template pool $\mathcal{P}$ and draws $\bar{\boldsymbol{\lambda}}^{(a_n)}$. The distribution combines normalized arm weights with uniform exploration, balancing previously rewarded allocations against alternative mixtures. Restricting selection to a compact pool replaces unrestricted search over the simplex of Tm loss components with explicit explore–exploit choices among interpretable templates. The selected template is ϵ-smoothed, mapped to logit space, and combined with the available bias $\mathbf{b}_n$ before softmax normalization, separating discrete template selection from continuous feedback refinement.

PID-based stabilization. At validation epoch e_n, the controller measures the tracking error $\varepsilon_n = \mathcal{J}_{e_n} - \mathcal{J}^\star$ and combines its instantaneous value, cumulative sum over controller rounds, and successive difference into the signal v_n. These terms provide proportional responsiveness, accumulated-error correction, and derivative damping when appropriately tuned. The signal updates the bias along the fixed directional prior $\mathbf{d}$ through Eq. 15, with componentwise clipping enforcing $\|\mathbf{b}_{n+1}\|_\infty \le B$. This bounds logit-space refinement and prevents unbounded bias drift. The updated bias affects the next allocation, rather than the training interval that generated the feedback.

Training under adaptive loss allocation. Before training, the availability mask $\mathbf{m}_n$ removes inactive task–modality components, and the remaining weights are renormalized. Provided the active set is nonempty, $\boldsymbol{\lambda}_n$ defines a valid simplex-weighted mixture over active losses. This allocation remains fixed throughout the round, including during minibatch differentiation, while the base optimizer performs its usual parameter updates with persistent optimizer state. Thus, BCLAN changes the relative contributions of task–modality gradients rather than the architecture or optimizer update rule, retaining compatibility with unimodal and multimodal learning pipelines.

Closed-loop feedback and theoretical perspective. At the end of each round, validation loss is evaluated under fixed uniform reference weights Λ^{ref}, preserving comparable feedback despite changing training allocations. The relative improvement since the preceding validation is converted into a bounded reward r_n, which updates only the selected arm using its sampling probability $p_n(a_n)$. A positive reward increases that arm's weight, while unplayed arm weights remain unchanged. Together with the updated PID bias, these weights determine the next allocation. The algorithm returns the checkpoint with the lowest fixed-reference loss among scheduled post-training validations.

The analysis establishes bounded multiplicative refinement around the selected template after smoothing, masking, and normalization (Lemma 1), together with a cumulative reward-transfer bound for the bandit update (Lemma 2). Under the stated assumptions on template progress, upward validation-loss variation, and round cost, Theorem 1 bounds the expected time to reach $\mathcal{J}^\star$. Corollary 1 further identifies when the progress advantage exceeds the cost of online adaptation relative to static weighting, accounting for validation and controller overhead.

Table 9: Regime-wise summary of BCLAN's dataset-level gains over the corresponding backbone baseline. The final two columns report positive pairs and pairs with gains of at least 0.5/1.0 points, respectively; #TTE denotes training time per epoch.

Regime	#Pairs	Mean gain ↑	Median gain ↑	Range	Mean gain (%) ↑	Mean #TTE red. (%) ↑	Positive pairs	≥0.5 / ≥1.0
Unpaired	36	0.87	0.83	0.10–2.70	1.09	40.9	36/36	34/7
Paired WSI+omics	9	0.81	0.70	0.50–1.40	1.01	46.0	9/9	9/2
Paired image–EHR/text	3	0.97	1.00	0.80–1.10	1.16	30.3	3/3	3/2
All	48	0.87	0.80	0.10–2.70	1.08	41.4	48/48	46/11

Table 10: Dataset-wise effect profile of BCLAN integrated into the corresponding backbone baseline, sorted by mean dataset-level gain.

Dataset	Regime	#Backbones	Mean gain ↑	Range	Mean gain (%) ↑
BRCA	Unpaired	6	1.43	0.80–2.70	2.51
CXR14	Paired image–EHR/text	1	1.10	1.10–1.10	1.36
GBMLGG	Paired WSI+omics	3	1.07	0.60–1.40	1.25
ADNI	Paired image–EHR/text	1	1.00	1.00–1.00	1.18
MORT	Unpaired	6	0.88	0.65–1.15	0.96
KVASIR	Unpaired	6	0.84	0.70–0.95	0.93
Open-I	Paired image–EHR/text	1	0.80	0.80–0.80	0.95
SIPaKMeD	Unpaired	6	0.80	0.55–0.95	0.84
CNMC-2019	Unpaired	6	0.75	0.60–0.85	0.78
KIRP	Paired WSI+omics	3	0.70	0.60–0.80	0.82
BLCA	Paired WSI+omics	3	0.67	0.50–0.80	0.95
HAM10000	Unpaired	6	0.52	0.10–0.90	0.54

D Dataset Descriptions

D.1 Datasets

We evaluate BCLAN on twelve heterogeneous medical datasets spanning unpaired imaging, unpaired omics and EHR, paired whole-slide image (WSI)–omics, and paired image–EHR and image–text settings. These benchmarks cover multi-disease classification, survival prediction, and mortality prediction. Table 6 summarizes their modalities, prediction tasks, and public scales; the latter describe the original releases rather than the task-specific filtered cohorts.

Unpaired imaging benchmarks. Following MAIL (Dhar et al., 2026c), we evaluate multi-disease classification using HAM10000 (Tschandl et al., 2018), SIPaKMeD (Plissiti et al., 2018), CNMC-2019 (Mourya et al., 2019), and KVASIR (Pogorelov et al., 2017). HAM10000 covers skin cancers, including melanoma and basal cell carcinoma, alongside benign and other pigmented lesions. SIPaKMeD supports cervical cancer screening through classification of normal, benign, and abnormal cervical cells, while CNMC-2019 distinguishes B-lineage acute lymphoblastic leukemia cells from normal B-lymphoid precursors. KVASIR covers gastrointestinal diseases and endoscopic findings, including esophagitis, polyps, and ulcerative colitis, together with anatomical landmarks and polyp-removal categories. These tasks assess adaptive optimization across diverse diseases and imaging modalities (Table 1).

Unpaired omics and EHR benchmarks. Following CURE (Dhar et al., 2026b), we evaluate TCGA-BRCA (Lingle et al., 2016) for survival prediction from omics data and MIMIC-III (MORT) (Johnson et al., 2016) for mortality prediction from intensive-care EHR. These benchmarks extend the evaluation to molecular and clinical measurements with distinct survival and mortality outcomes (Table 1).

Paired WSI–omics benchmarks. Following the survival-prediction setup of MMLego (Hemker et al., 2025), we evaluate BCLAN on TCGA-BLCA (Network et al., 2014), GBMLGG (Bakas et al., 2017), and TCGA-KIRP (Linehan et al., 2016). Histopathology slides are paired with patient-matched molecular profiles, including gene expression, copy-number variation, and mutation data, providing complementary morphological and molecular information for survival prediction. Survival duration and censoring status provide patient-level supervision, and predictive performance is evaluated using C-Index (Table 2).

Paired image–EHR and image–text benchmarks. We evaluate classification on ADNI (Jack Jr et al., 2008), CXR14 (Wang et al., 2017), and Open-I (Demner-Fushman et al., 2016). ADNI combines neuroimaging with clinical variables for disease classification, adapting MAIL's image preprocessing to paired image–EHR inputs (Dhar et al., 2026c). For CXR14 and Open-I, the image–text formulation of CARZero (Lai et al., 2024) provides the reference for multi-label chest-disease classification using radiographs and textual disease descriptions (Table 3).

D.2 Dataset Preprocessing Details.

Preprocessing follows the following principles.

Unpaired imaging inputs and data partitioning. Following MAIL (Dhar et al., 2026c), classification images are resized to $128\times128\times3$. HAM10000, SIPaKMeD, and KVASIR use $80\%/10\%/10\%$ training/validation/test partitions, whereas CNMC-2019 retains its predefined partitions. In our implementation, grayscale inputs are converted to three channels, pixel intensities are normalized to $[0, 1]$, and patient-level separation is maintained whenever patient identifiers are available.

Unpaired omics and EHR preprocessing. Following CURE (Dhar et al., 2026b), non-imaging vectors are converted into pseudo-images for convolutional processing. In our implementation, features are harmonized within each cohort and standardized using z-score statistics computed exclusively from the training split. Each vector is mapped to a single-channel 128×128 grid using bicubic interpolation and replicated across three channels. BRCA and MORT use $80\%/10\%/10\%$ training/validation/test splits, with random upsampling restricted to the training split to keep effective training-sample counts comparable across datasets. The survival and mortality labels remain unchanged by these input transformations.

Table 11: Hyperparameter settings of multimodal fusion integrated with BCLAN.

Setting	Value
Training epochs	200
Optimizer	Adam
Initial learning rate	1×10^{-3}
Scheduler	ReduceLROnPlateau
Minimum learning rate	1×10^{-6}
Survival-task loss	Negative log-likelihood
Classification loss	Cross-entropy
Hardware	1× NVIDIA A100 80GB GPU
Operating system	Ubuntu
Repeated runs	5

Paired WSI–omics preprocessing. Following MMLego (Hemker et al., 2025), each slide is converted into a bag of 256×256 RGB patches. A ResNet50 pretrained on Kather100K encodes each patch into a 2048-dimensional feature vector. Non-variable mutation and copy-number variation features are removed, while low-variability gene-expression features are filtered and retained expression values undergo a $\log(1+x)$ transformation. Omics remain one-dimensional tabular inputs, paired with histopathology features and patient-level survival labels. We follow the repeated random-subsampling protocol with five $70\%/15\%/15\%$ training/validation/test splits, keeping all records from each patient within the same partition.

Paired image–EHR preprocessing. We follow MAIL's image preprocessing (Dhar et al., 2026c), resizing imaging inputs to $128 \times 128 \times 3$, and adapt the pipeline to incorporate paired clinical measurements. In our implementation, image intensities are normalized to $[0, 1]$, while EHR features are z-score standardized using training-split statistics. Each clinical vector is mapped to a single-channel 128×128 grid using bicubic interpolation and replicated across three channels. Images and their associated clinical records remain in the same patient partition, preserving paired supervision for disease classification.

Paired image–text preprocessing. Following CARZero (Lai et al., 2024), chest radiographs are resized to 224×224. During image–report training, text from the findings and impression sections is segmented into sentences, with one sentence sampled at each training iteration. Disease-specific textual prompts provide the language inputs for multi-label classification. The CXR14 evaluation uses the official test partition, following the reference protocol.

D.3 Data Augmentation Protocols.

For unpaired imaging and the adapted image–EHR pipeline, we follow MAIL (Dhar et al., 2026c): each training image produces three additional variants through a 20° rotation, a 5-pixel translation along both spatial axes, and Gaussian smoothing with a 3×3 kernel. Augmented images retain their

labels and, where applicable, their associated clinical records. For image–text training, we follow CARZero (Lai et al., 2024) using random horizontal flips, affine transformations, and color jittering. Augmentation is confined to training images; validation and test images are not augmented.

E Implementation Details

Benchmarks and preprocessing. We evaluate multi-disease classification, survival prediction, and mortality prediction across the twelve datasets in App. D. Unpaired imaging follows MAIL (Dhar et al., 2026c), while paired image–EHR classification adapts its image preprocessing. Unpaired omics and EHR follow CURE (Dhar et al., 2026b); paired WSI–omics follows the survival-prediction setup and preprocessing of MMLego (Hemker et al., 2025); and image–text classification follows CARZero (Lai et al., 2024). Within each host–dataset comparison, preprocessing and data partitions remain fixed across the unmodified pipeline and its online-adaptation variants.

Compared methods and backbones. We compare BCLAN with ADAMBS (Liu et al., 2020), MABSampler (Ganapathiraman et al., 2023), Incremental PID (Wang & Zhang, 2022), and PID-AGD (Zhang & Xie, 2024) in the configurations reported in Tables 1–3. The seven host baselines comprise the unimodal learner NAT (Hassani et al., 2023); the vision–language model CARZero (Lai et al., 2024); and the multimodal fusion models HEALNet (Hemker et al., 2024), DRIFA-Net (Dhar et al., 2025), HyPCA-Net (Dhar et al., 2026a), MMLego (Hemker et al., 2025), and MAIL (Dhar et al., 2026c). Each comparison uses the unmodified host and its available online-adaptation variants.

Optimization protocol and task-specific losses. All compared pipelines are trained for 200 epochs on 1× NVIDIA A100 80GB GPU under Ubuntu. Optimization uses Adam with initial learning rate 1×10^{-3} and ReduceLROnPlateau with minimum learning rate 1×10^{-6}. Survival prediction from omics or paired WSI–omics uses negative log-likelihood (NLL) loss, following (Hemker et al., 2024; 2025). Supervised disease and mortality classification from imaging and EHR uses cross-entropy loss, following (Dhar et al., 2026c;b). BCLAN changes the relative contributions of the controlled loss components, rather than replacing their loss functions. Common training settings are summarized in Table 11.

Evaluation criteria and repeated runs. We report accuracy (ACC), AUC, and concordance index (C-Index), together with mean average precision (mAP) for ADNI in Table 3. Computational measures comprise parameter count (#P, in millions), floating-point operations (#F, in GFLOPs), and training time per epoch (#TTE, in seconds). The per-epoch measure #TTE is distinct from cumulative validation-loss convergence time $\mathcal{T}^\star$ in Eq. 4. Results are averaged over 5 random seeds, with standard deviations included in the corresponding comparisons; bold denotes the best result in each comparison.

Controller integration and update schedule. BCLAN augments the base optimizer without modifying the host architecture, trainable backbone parameter count, or backbone FLOPs. The selected template and current bias determine the training allocation through the logit-space construction and masked normalization in Eqs. 12 and 7. This allocation remains fixed during minibatch differentiation, while the base optimizer retains its state across updates. We index controller updates by validation round n, ending at epoch e_n, rather than by every training epoch. The score-based configuration below uses a validation interval of $\Delta_v = 2$ epochs; the resulting feedback updates the arm weights and bias for the next interval. Fixed-reference validation loss $\mathcal{J}_{e_n}$ is monitored under uniform weights Λ^{ref}, separately from the score-based reward defined below.

Reward and controller instantiation. We specify the two-head, score-based configuration using the validation accuracies $a^{\mathrm{val}}_{s,e_n}, a^{\mathrm{val}}_{h,e_n} \in [0,1]$ of the logged heads task_s and task_h. With reward_reduction=mean, the reward is

$$r_n^{\mathrm{sc}} = \begin{cases} \mathrm{clip}\left(\dfrac{a^{\mathrm{val}}_{s,e_n} + a^{\mathrm{val}}_{h,e_n}}{2},\, 0,\, 1\right), & \text{if both accuracies are available,} \\ \dfrac{1}{1+\mathcal{L}^{\mathrm{val}}_{e_n}}, & \text{otherwise,} \end{cases} \tag{42}$$

where $\mathcal{L}_{e_n}^{\text{val}}$ is the validation loss provided by the callback logs. The fallback reward is bounded for finite, nonnegative validation losses. Unlike the relative loss-improvement reward in Eq. 10, r_n^{sc} measures the current validation-score level.

Score tracking and integral clipping. The controller tracks a fixed score target $s^\star = 0.88$:

$$\varepsilon_n = s^\star - r_n^{\text{sc}}. \tag{43}$$

The `PID` signal is

$$v_n = K_p\varepsilon_n + K_i \operatorname{clip}\left(\sum_{j=1}^{n}\varepsilon_j,\, -50,\, 50\right) + K_d(\varepsilon_n - \varepsilon_{n-1}), \tag{44}$$

with $(K_p, K_i, K_d) = (2.0, 0.4, 0.2)$. The sum and successive difference are computed over controller updates, and ε_0 denotes the callback's initial stored tracking error. Clipping bounds the accumulated error entering the integral term; it is separate from the componentwise bias clipping below.

Directional bias refinement. For available head accuracies, define

$$\mathbf{z}_n = \begin{bmatrix} a_{h,e_n}^{\text{val}} - a_{s,e_n}^{\text{val}} \\ a_{s,e_n}^{\text{val}} - a_{h,e_n}^{\text{val}} \end{bmatrix}. \tag{45}$$

The refinement direction is

$$\mathbf{d}_n = \begin{cases} \mathbf{z}_n/\|\mathbf{z}_n\|_2, & \text{if both accuracies are available and } \|\mathbf{z}_n\|_2 \geq 10^{-8}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{46}$$

The bias update is

$$\mathbf{b}_{n+1} = \operatorname{clip}(\mathbf{b}_n + \eta_{\text{ctrl}} v_n \mathbf{d}_n,\, -B,\, B)\,, \qquad \eta_{\text{ctrl}} = 0.20, \quad B = 2.0. \tag{47}$$

Before clipping, the proposed increment favors the lower-accuracy head when $v_n > 0$ and reverses direction when $v_n < 0$. When $\mathbf{d}_n = \mathbf{0}$, a bias already within the clipping bounds remains unchanged. Consequently, missing head accuracies activate the loss-based reward fallback but not directional `PID` refinement; the bandit can still update from the fallback reward.

Template pool and bandit update. For this two-component configuration, the template pool contains $A = 5$ allocations, with entries ordered as $(\texttt{task_s}, \texttt{task_h})$:

$$\mathcal{P}_2 = \{(0.50, 0.50), (0.65, 0.35), (0.35, 0.65), (0.80, 0.20), (0.20, 0.80)\}. \tag{48}$$

The initial template distribution is uniform. Using arm weights $w_n(a)$ and exploration rate $\gamma = 0.07$, we sample a_n from

$$p_n(a) = (1-\gamma)\frac{w_n(a)}{\sum_{b=1}^{A} w_n(b)} + \frac{\gamma}{A}, \qquad a_n \sim p_n. \tag{49}$$

The played-arm reward estimate and update are

$$\begin{aligned} \widehat{r}_n &= \frac{r_n^{\text{sc}}}{p_n(a_n) + \xi + \epsilon_{\text{IX}}}, \\ \widetilde{w}_{n+1}(a) &= \begin{cases} w_n(a)\exp(\eta\widehat{r}_n), & a = a_n, \\ w_n(a), & a \neq a_n, \end{cases} \\ w_{n+1}(a) &= \frac{\widetilde{w}_{n+1}(a)}{\sum_{b=1}^{A}\widetilde{w}_{n+1}(b)}, \end{aligned} \tag{50}$$

where $\xi = 0.05$ and $\epsilon_{\text{IX}} = 10^{-6}$. The effective bandit learning rate is $\eta = 0.5/A = 0.1$, corresponding to the implementation's $0.5/K$ factor with $K = 5$. Only the played arm receives the multiplicative reward update; all arm weights are subsequently normalized. Table 7 summarizes this score-based configuration.

Scope of the controller specification. The two-head score-based configuration above is distinct from the loss-based controller in §. 2.3 and Alg. 1: it uses score-level rewards, a score target, a feedback-dependent direction, and clipping of the accumulated error entering the integral term. Theorem 1 and Corollary 1 concern the loss-based configuration and do not, by themselves, establish validation-loss convergence for this score-based instantiation. The two-component pool and direction specified here do not define a generalization to arbitrary Tm-component allocations.

F Additional Experimental Results

To complement the loss-landscape visualizations in Figs. 4 and 11, we further analyze our work as different way of perspectives as shown in the following.

Rank-based aggregation and family-wise dominance. To aggregate comparisons across ACC, AUC, C−Index, and mAP, Fig. 6 ranks methods within each reported comparison cell and averages their ranks over available cells. BCLAN achieves the best mean performance rank (1.02) and training-time rank (1.00). The corresponding runners-up are PID−AGD (2.33) for predictive performance and ADAMBS (2.20) for #TTE, highlighting that faster training alone does not ensure comparable predictive performance.

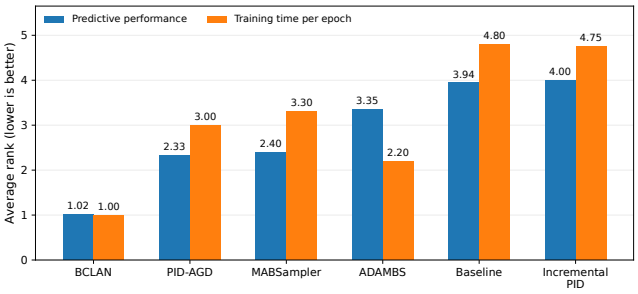

Figure 6: Average rank across all reported comparisons. Lower rank is better. Performance ranks are computed over all metric cells reported in Tables 1–3, whereas TTE ranks are computed backbone-wise over the methods available for that backbone. BCLAN ranks first in predictive performance and in training time efficiency metrics.

Table 8 further shows that these gains extend across competing adaptation families. Compared with the corresponding unmodified pipelines and their ADAMBS and MABSampler variants, BCLAN improves 77 of 78 comparable performance cells and ties on the remaining cell; it improves all 12 comparable cells against each of Incremental PID and PID−AGD. Dataset-level averaging yields positive gains across all 48 backbone–dataset pairs. At the backbone level, BCLAN dominates all 38 compared alternatives in the two-dimensional performance–training-time space, achieving higher overall performance and lower #TTE.

Regime-wise heterogeneity and non-outlier behavior. Table 9 summarizes dataset-level gains over the corresponding backbone baselines, averaging the metric pair for datasets with two reported metrics. All 48 backbone–dataset pairs improve, with 46 gaining at least 0.5 points and 11 gaining at least 1.0 point. The mean gain is 0.87 points (median 0.80), corresponding to a mean relative improvement of 1.08%. The largest mean performance gain occurs on paired image–EHR/text benchmarks (0.97 points), whereas paired WSI+omics yields the largest mean #TTE reduction (46.0%; Fig. 7).

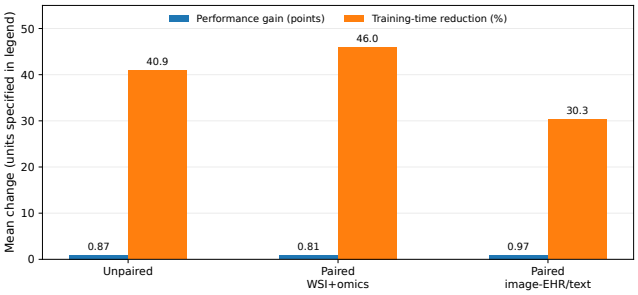

Figure 7: **Regime-wise gains over the corresponding backbone.** Bars report the mean predictive-performance gain in points and the mean #TTE reduction in percent for unpaired, paired WSI–omics, and paired image–EHR/text settings.

Dataset-wise effect profile. Table 10 and Fig. 8 provide a finer-grained view across the 12 benchmarks. The strongest average gain is obtained on BRCA (mean +1.43, range +0.80 to +2.70 across six backbones), followed by CXR14 (+1.10), GBMLGG (+1.07), and ADNI (+1.00). Even the smallest average gain, on HAM10000 (+0.53 on average), remains positive. This pattern suggests that BCLAN is especially useful when the optimization landscape is more heterogeneous or cross-modal, while remaining safe on already strong backbones and saturated datasets.

Backbone-wise efficiency–performance frontier. Tables 12–13 and Fig. 9 show consistent performance–training-time improvements across the ten evaluated backbone–regime configurations. Relative to the corresponding unmodified pipelines, BCLAN improves the overall score by 0.57–1.00 points and reduces #TTE by 30.3%–50.4%. The largest training-time reduction occurs for DRIFA−Net on paired WSI+omics tasks (50.4%). Against the strongest non-BCLAN alternative

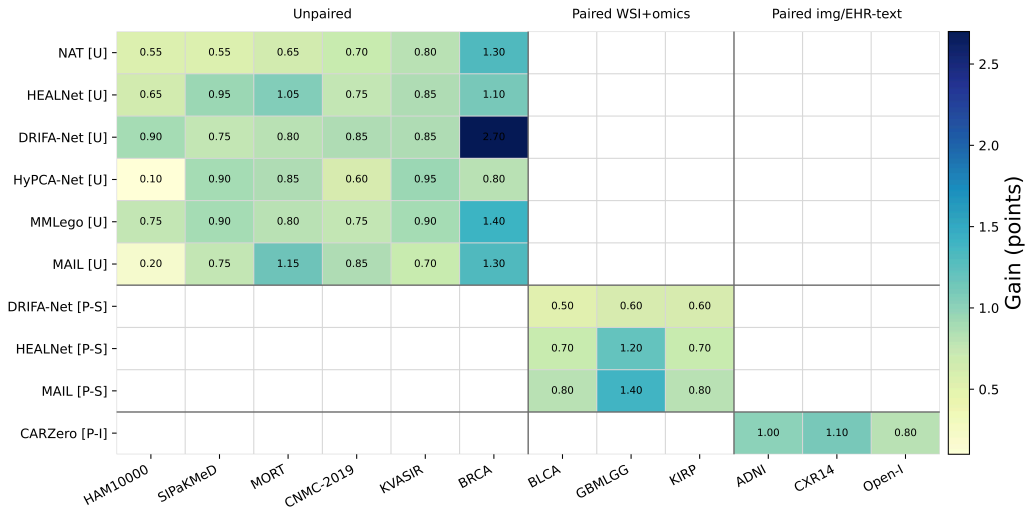

Figure 8: Dataset-wise performance gain of BCLAN over the corresponding backbone baseline. Each cell is a dataset-level score obtained by averaging all metrics reported for that dataset within a backbone block. Gains are positive for all 48 backbone–dataset pairs. Here, U denotes unpaired and P−S denotes paired WSI + Omics and P−I denotes paired image + EHR/texts.

Table 12: Backbone-wise gains of BCLAN integrated into the corresponding backbone baseline. "Overall" denotes the mean of all performance metrics reported for that backbone block. Abbreviations: U = unpaired, P-S = paired WSI+omics, and P-I = paired image–EHR/text.

Backbone	Δ Overall ↑	gain (%) ↑	#TTE red. (%) ↑
Unpaired			
NAT (U)	0.71	0.81	46.7
HEALNet (U)	0.87	0.96	42.8
DRIFA-Net (U)	1.00	1.09	43.6
HyPCA-Net (U)	0.69	0.75	38.9
MMLego (U)	0.87	0.96	35.6
MAIL (U)	0.78	0.85	38.0
Paired WSI+omics			
DRIFA-Net (P-S)	0.57	0.72	50.4
HEALNet (P-S)	0.87	1.07	48.2
MAIL (P-S)	1.00	1.24	39.4
Paired image–EHR/text			
CARZero (P-I)	0.97	1.16	30.3

in each configuration, selected by overall performance with ties broken by lower #TTE, BCLAN retains gains of 0.43–0.77 points with further #TTE reductions of 10.3%–36.9%, supporting a favorable training-time efficiency–performance trade-off.

F.1 Additional BCLAN Convergence Analysis

Conventional multimodal fusion backbones such as MAIL (Dhar et al., 2026c) are typically optimised in an *open-loop* manner with a fixed multi-task loss mixture, so the loss allocation cannot adapt when optimisation dynamics drift as shared representations evolve across modalities (ref. Figs. 10–11). BCLAN accelerates convergence by transforming this static setup into a *closed-loop*, validation-driven loss-reweighting process (ref. Figs. 10–11). Concretely, it performs a coarse explore–exploit search over a small pool of loss-weight templates via an adversarial bandit, then applies PID-based refinement in logit space to stabilise and fine-tune the selected allocation under noisy validation feedback. Because BCLAN augments the base optimiser rather than modifying the

Table 13: Backbone-wise gains of BCLAN over the strongest non-BCLAN alternative, selected by overall performance with ties broken by lower #TTE.

Backbone	Strongest alt.	Δ Overall ↑	#TTE red. (%) ↑
Unpaired			
NAT (U)	MABSampler	0.58	30.9
HEALNet (U)	MABSampler	0.61	36.9
DRIFA-Net (U)	MABSampler	0.77	30.3
HyPCA-Net (U)	MABSampler	0.55	25.1
MMLego (U)	MABSampler	0.62	20.6
MAIL (U)	MABSampler	0.56	25.0
Paired WSI+omics			
DRIFA-Net (P-S)	PID-AGD	0.43	23.7
HEALNet (P-S)	MABSampler	0.50	20.1
MAIL (P-S)	PID-AGD	0.50	19.7
Paired image–EHR/text			
CARZero (P-I)	PID-AGD	0.47	10.3

Table 14: Fine-grained ablation of the PID refinement while keeping the bandit aware EXP3-IX template selector active. "Overall score" is the average of the eight ACC/AUC entries across HAM10000, SIPaKMeD, MORT, and CNMC-2019. $TV(\lambda)$ denotes the total variation of the outer-loop loss-weight trajectory, and "Switch rate" is the fraction of validation updates that change the dominant template.

Variant	Overall score ↑	#TTE ↓	$TV(\lambda)$ ↓	Switch rate ↓
Bandit + P	97.02	18.9	0.214	0.38
Bandit + PI	97.17	17.7	0.178	0.30
Bandit + PD	97.12	18.0	0.146	0.23
Bandit + PID	**97.29**	**16.8**	**0.121**	**0.18**

fusion backbone, it can be integrated into existing multimodal fusion architectures such as MAIL without architectural changes.

In the core-component ablation, bandit exploration improves training-time efficiency through coarse allocation search, while PID refinement damps noisy reweighting and corrects drift; their combination achieves the strongest performance–training-time trade-off. In the broader multimodal comparison, integrating BCLAN with MAIL improves MORT AUC from 94.4 to 95.6 and KVASIR AUC from 92.3 to 93.2, while reducing #TTE from 27.1 to 16.8 s, demonstrating improved predictive performance and training-time efficiency.

G Additional Ablation Study

To disentangle how each controller component drives BCLAN's efficiency–accuracy trade–off, we conduct three complementary ablations on the representative MAIL-based *unpaired* setting evaluated on four benchmarks: HAM10000, SIPaKMeD, MORT, and CNMC-2019.

1. **PID-term decomposition.** To pinpoint which branches contribute most to training stability, we ablate the proportional (P), integral (I), and derivative (D) terms individually and in combination.
2. **Template-pool sweep.** We vary the number of loss-weight templates to characterise the exploration–approximation trade-off under a finite controller budget.
3. **Validation-interval/overhead analysis.** By modulating the outer-loop validation cadence and explicitly measuring wall-clock controller overhead, we test whether the efficiency gains persist once this cost is accounted for.

Across all three perspectives, a consistent picture emerges: *only the structured interplay between coarse bandit exploration and low-variance closed-loop PID refinement yields the observed gains*. No single heuristic reproduces BCLAN's stability–efficiency advantage in isolation.

G.1 Fine-grained PID-term Ablation.

Table 14 provides a finer decomposition into Proportional-only (P), Proportional with Integral (PI), Proportional with Derivative (PD), and Proportional with Integral and Derivative (PID), while keeping the bandit component active throughout. The results reveal a clean division of labor across the

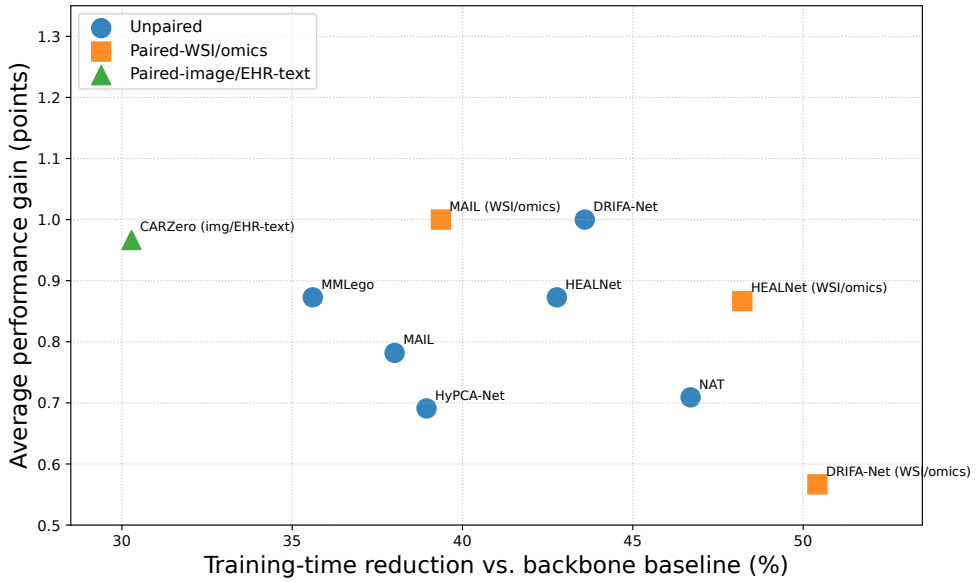

Figure 9: Backbone-level efficiency–performance gains of BCLAN integrated into the corresponding backbone baseline. Every point lies in the desirable quadrant with higher performance and lower training time.

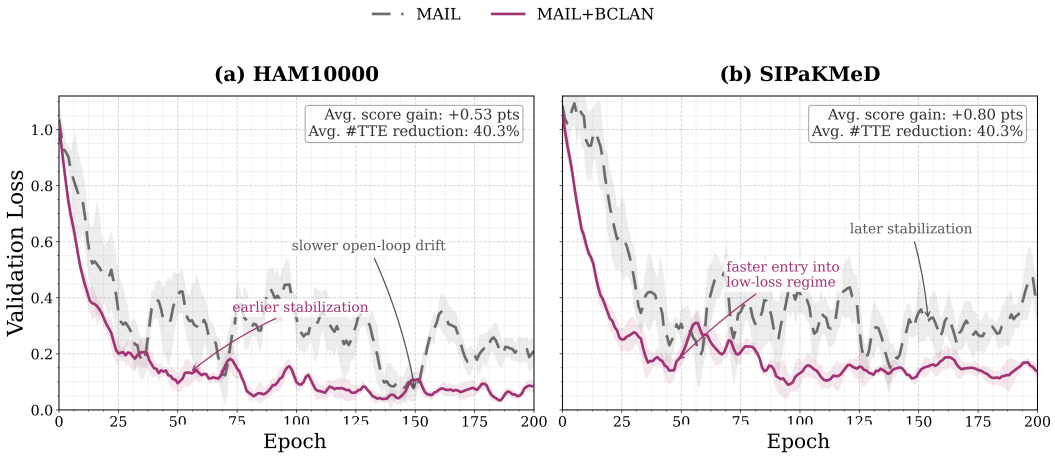

Figure 10: BCLAN: closed-loop loss reweighting for accelerated convergence on unpaired modality datasets.

PID terms. A proportional controller already yields a clear gain, improving the overall score to 97.02 and reducing #TTE to 18.9, which indicates that the validation-driven feedback signal is informative even under first-order tracking alone. Adding the integral term further improves the score to 97.17 and lowers #TTE to 17.7, suggesting that cumulative error correction is useful for removing the steady-state bias of the coarse bandit-selected template. By contrast, adding the derivative term has its most visible effect on stability: with respect to Bandit + P, Bandit + PD reduces the total variation of the loss-weight trajectory from 0.214 to 0.146 and lowers the template-switching rate from 0.38 to 0.23. The overall PID controller combines these benefits and delivers the strongest overall trade-off, reaching the best overall score (97.29), the lowest training cost (16.8 #TTE), and the smoothest outer-loop dynamics.

This pattern is important because it shows that BCLAN does not improve performance by merely reacting more aggressively to noisy validation fluctuations. Instead, the integral term removes persistent allocation bias, while the derivative term damps over-reaction to short-horizon noise. In other words, the bandit proposes a promising coarse direction, and the overall PID loop is what turns that direction into smooth and sustained optimization progress.

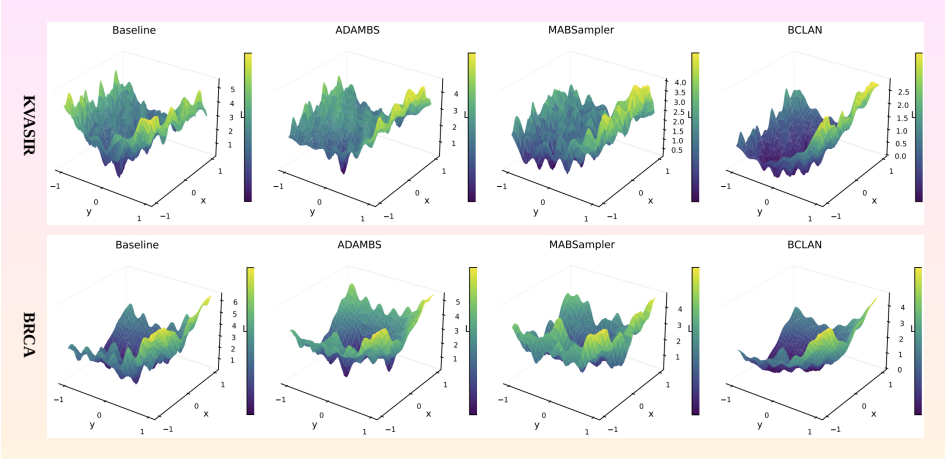

Figure 11: **Loss-landscape comparison on unpaired benchmarks.** `3D` visualizations of the (validation) loss surface in a `2D` neighborhood of the converged solution for representative unpaired heterogeneous-modality datasets. The horizontal axes (x, y) parameterize displacements along two normalized directions in parameter space, and the vertical axis denotes the resulting loss. Columns compare the baseline optimizer and prior online adaptive methods (`ADAMBS`, `MABSampler`) against `BCLAN`. Across datasets, `BCLAN` converges to broader, flatter low-loss basins, indicating more stable optimization dynamics under heterogeneous feedback.

Table 15: Effect of the template-pool size A. "Uniform fixed" uses a static uniform loss mixture. "Oracle fixed" is the best single template in hindsight from the same pool. "Gap closed" is computed as (`BCLAN` − Uniform)/(Oracle − Uniform). "Arm entropy" denotes the average entropy of the `EXP3-IX` sampling distribution across outer-loop updates.

A	Uniform fixed	Oracle fixed	BCLAN	Gap closed ↑	#TTE ↓	Arm entropy
2	96.55	96.87	96.79	75.0%	18.6	0.38
4	96.55	97.09	97.01	85.2%	17.4	0.61
8	96.55	97.35	**97.29**	**92.5%**	**16.8**	0.84
16	96.55	**97.38**	97.20	78.3%	17.7	1.08

G.2 Effect of Template-pool Size.

Table 15 studies the role of the template-pool size A and compares `BCLAN` against the best fixed template selected in hindsight from the same pool. The results reveal a clear finite-budget trade-off. When the pool is too small ($A = 2$), the controller closes only 75.0% of the uniform-to-oracle gap and the overall score remains limited to 96.79, indicating that a very coarse discretization undercovers the useful region of the allocation space. Increasing the pool to $A = 4$ and then $A = 8$ substantially improves both accuracy and efficiency, with `BCLAN` reaching 97.29 and closing 92.5% of the oracle gap at $A = 8$. However, pushing the pool size further to $A = 16$ no longer helps in practice: although the oracle fixed-template score improves marginally from 97.35 to 97.38, the online `BCLAN` score decreases to 97.20 and `#TTE` rises to 17.7.

This separation between the oracle and online curves is precisely the phenomenon predicted by the finite-horizon theory. A richer pool improves representational coverage, but it also enlarges the exploration burden that the controller must resolve within a fixed training budget. The increase in average arm entropy from 0.38 to 1.08 further supports this interpretation, as it indicates that exploration remains more diffuse in larger pools. Overall, the empirical optimum at $A = 8$ is consistent with the expected trade-off between discretization quality and the $O(\sqrt{A \log A/E})$ cost of online exploration. Equally importantly, `BCLAN` remains close to the best fixed template across all pool sizes, which supports the claim that the bandit layer approaches the strongest template in hindsight while retaining the flexibility of online adaptation.

Table 16: Effect of the outer-loop validation interval Δ_v. "Ctrl. time" measures the average controller wall-clock cost per training epoch. "Overhead" is the percentage of total epoch time attributable to the controller. $|\widehat{T}_R^\star - T^\star|$ measures the discrepancy between the reward-based target-attainment estimate and the observed attainment epoch.

Δ_v	Overall score ↑	#TTE ↓	Ctrl. time / epoch ↓	Overhead ↓	$\lvert\widehat{T}_R^\star - T^\star\rvert$ ↓	Peak GPU mem.
1	97.23	17.1	0.41 s	2.4%	1.2	13.4 GB
2	**97.29**	**16.8**	0.24 s	1.4%	**0.8**	13.3 GB
5	97.18	16.9	0.12 s	0.7%	1.4	13.3 GB
10	96.97	17.6	**0.07 s**	**0.4%**	2.7	**13.2 GB**

G.3 Validation Interval and Controller Overhead.

Table 16 varies the outer-loop validation interval Δ_v and reports both effectiveness and explicit controller cost. The main conclusion is that BCLAN is robust to a broad range of update cadences, but the best trade-off is obtained at an intermediate interval. Updating every epoch ($\Delta_v = 1$) yields strong performance, yet slightly underperforms $\Delta_v = 2$ despite incurring higher controller cost (0.41 s/epoch versus 0.24 s/epoch). This suggests that maximal responsiveness is not necessarily optimal, likely because very frequent updates expose the controller to noisier short-term validation fluctuations. At the opposite extreme, infrequent updates ($\Delta_v = 10$) reduce the controller overhead to only 0.4%, but they also noticeably degrade both the overall score (96.97) and target-attainment efficiency (17.6 #TTE), indicating that stale feedback weakens the ability of the outer loop to track non-stationary training dynamics.

The intermediate cadence $\Delta_v = 2$ provides the best balance of responsiveness and stability, yielding the strongest overall score (97.29), the best efficiency (16.8 #TTE), and still negligible overhead (1.4%). Importantly, the reward-certificate diagnostic remains tight throughout the sweep, with $|\widehat{T}_R^\star - T^\star|$ staying below one epoch at the best setting and small overall, which gives empirical support to the theoretical claim that cumulative reward is not merely a surrogate training signal but a practical certificate of target attainment. Peak GPU memory is essentially unchanged across the sweep, as expected, since BCLAN introduces no extra trainable backbone parameters and only modifies loss reweighting at the outer loop.

Hence, these analyses substantiate our central claim: the bandit module furnishes a principled explore–exploit strategy for selecting loss-weight templates, while the PID regulator translates this coarse search into smooth, low-variance updates. Crucially, the overall controller remains lightweight, so its computational overhead never offsets the optimization-efficiency gains it delivers.

H Discussions

Our results suggest that multimodal fusion is bottlenecked not only by representational expressivity, but also by how optimization effort is allocated across heterogeneous task-modality losses over time. In medical settings, shared representations evolve unevenly across imaging, omics, EHR, and text, so the relative usefulness of each loss component is inherently non-stationary throughout training. A fixed open-loop loss mixture is therefore unlikely to remain aligned with the optimization dynamics across the full trajectory. From this perspective, slow convergence is not simply a property of the backbone or the base optimizer; it reflects a control mismatch between a drifting learning system and a static allocation rule.

BCLAN addresses this mismatch by recasting loss weighting as a closed-loop control problem. Its advantage arises from a deliberate decomposition of adaptation into coarse search and fine stabilization. The adversarial bandit searches over a compact pool of simplex-constrained templates and provides an explicit explore-exploit mechanism for discovering allocations that yield earlier validation improvement. The PID module then refines the selected template in logit space, suppressing abrupt reweighting, correcting persistent bias, and damping sensitivity to noisy feedback. Equally important, we evaluate progress under a fixed reference weighting, so the outer loop receives a comparable validation signal even when training-time allocations change. Together with masking and renormalization over active terms, this allows the same controller to operate in both paired and unpaired heterogeneous regimes.

Algorithm 1: BCLAN: validation-driven loss reweighting

Input: training and validation data $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}$; initial parameters β_0;
base optimizer OptStep with state s_0 and learning rates $\{\eta_e\}_{e=1}^{E}$;
validation schedule $0 = e_0 < e_1 < \cdots < e_N = E$;
template pool $\mathcal{P} = \{\bar{\boldsymbol{\lambda}}^{(a)}\}_{a=1}^{A} \subset \Delta^{Tm-1}$;
fixed uniform reference weights Λ^{ref}; target loss $\mathcal{J}^\star$; smoothing $\epsilon > 0$;
prescribed masks $\{\mathbf{m}_n \in \{0,1\}^{Tm}\}_{n=1}^{N}$ with $\|\mathbf{m}_n\|_1 > 0$;
bandit parameters $\eta, \xi > 0$, $\gamma \in [0,1]$;
PID parameters $(K_p, K_i, K_d, \lambda_b, B, \mathbf{d})$, with $\lambda_b, B > 0$ and $\mathbf{d} \in \mathbb{R}^{Tm}$.
Output: best post-training validation checkpoint $\beta^\star$

$\beta \leftarrow \beta_0$, $s \leftarrow s_0$, $\mathbf{u}_1 \leftarrow \mathbf{1}$, $\mathbf{b}_1 \leftarrow \mathbf{0}$
$\mathcal{J}_0 \leftarrow \mathcal{L}_{\mathtt{MTL}}^{\mathrm{val}}(\beta_0; \Lambda^{\mathrm{ref}})$
$\varepsilon_0 \leftarrow \mathcal{J}_0 - \mathcal{J}^\star$, $I_0 \leftarrow 0$, $\bar{\mathcal{J}} \leftarrow +\infty$
for $n = 1$ **to** N **do**
  $p_n(a) \leftarrow (1-\gamma)\dfrac{\mathbf{u}_n(a)}{\|\mathbf{u}_n\|_1} + \dfrac{\gamma}{A}, \quad a = 1, \ldots, A$
  Sample $a_n \sim p_n$ ▷ EXP3-IX-style template selection
  $\boldsymbol{\theta}_n \leftarrow \log\Big(\bar{\boldsymbol{\lambda}}^{(a_n)} + \epsilon\mathbf{1}\Big) + \mathbf{b}_n$
  $\tilde{\boldsymbol{\lambda}}_n \leftarrow \mathbf{m}_n \odot \mathrm{softmax}(\boldsymbol{\theta}_n)$
  $\boldsymbol{\lambda}_n \leftarrow \tilde{\boldsymbol{\lambda}}_n / \|\tilde{\boldsymbol{\lambda}}_n\|_1$ ▷ normalize over active losses
  for $e = e_{n-1} + 1$ **to** e_n **do**
    foreach *minibatch* $\mathcal{B}$ *from* $\mathcal{D}_{\mathrm{tr}}$ *in epoch* e **do**
      $\mathbf{g} \leftarrow \sum_{j:\, m_{n,j}=1} \lambda_{n,j} \nabla_\beta L_j(\beta; \mathcal{B})$ ▷ hold $\boldsymbol{\lambda}_n$ fixed
      $(\beta, s) \leftarrow \mathrm{OptStep}(\beta, s, \mathbf{g}; \eta_e)$
    $\beta_e \leftarrow \beta$
  $\mathcal{J}_{e_n} \leftarrow \mathcal{L}_{\mathtt{MTL}}^{\mathrm{val}}(\beta_{e_n}; \Lambda^{\mathrm{ref}})$
  $r_n \leftarrow \mathrm{clip}\left(\dfrac{\mathcal{J}_{e_{n-1}} - \mathcal{J}_{e_n}}{|\mathcal{J}_{e_{n-1}}| + \epsilon},\, 0,\, 1\right)$
  $\mathbf{u}_{n+1} \leftarrow \mathbf{u}_n$
  $\mathbf{u}_{n+1}(a_n) \leftarrow \mathbf{u}_n(a_n) \exp\left(\eta\, \dfrac{r_n}{p_n(a_n) + \xi}\right)$ ▷ update only the played arm
  $\varepsilon_n \leftarrow \mathcal{J}_{e_n} - \mathcal{J}^\star, \quad I_n \leftarrow I_{n-1} + \varepsilon_n$
  $v_n \leftarrow K_p\varepsilon_n + K_i I_n + K_d(\varepsilon_n - \varepsilon_{n-1})$
  $\mathbf{b}_{n+1} \leftarrow \mathrm{clip}(\mathbf{b}_n - \lambda_b v_n \mathbf{d},\, -B,\, B)$ ▷ bounded bias refinement
  if $\mathcal{J}_{e_n} < \bar{\mathcal{J}}$ **then**
    $\bar{\mathcal{J}} \leftarrow \mathcal{J}_{e_n}, \quad \beta^\star \leftarrow \mathrm{copy}(\beta_{e_n})$
return $\beta^\star$

A useful way to interpret BCLAN is as an optimizer-side mechanism for reallocating gradient budget rather than an architecture-side mechanism for adding capacity. Because the backbone, parameter count, and FLOP budget remain unchanged, the gains cannot be attributed to a more expressive model. Instead, BCLAN changes which task-modality components dominate the shared update at different stages of learning. This is especially important in multimodal fusion, where dominant modalities can suppress weaker ones and where different modalities often become informative at different rates. By adapting the loss mixture online, BCLAN reduces this mismatch and moves the model earlier into a validation regime that generalizes well under a finite compute budget.

Our theoretical analysis clarifies the precise sense in which this efficiency-performance trade-off is optimal. We do not claim unrestricted global optimality over all possible continuous weighting rules. Rather, within the template class, the EXP3−IX outer loop achieves minimax-optimal explore-exploit efficiency up to the standard sublinear regret term, while the PID refinement keeps the realized allocation within a bounded neighborhood of the sampled template through anti-windup clipping and simplex-preserving logit refinement. This combination is critical: the bandit gives adaptivity to drifting optima, whereas the controller ensures that exploration does not translate into unstable loss-weight oscillations. The complementary approximation-estimation decomposi-

tion further separates template discretization error, bounded PID deviation, and online learning error, while the reward-certificate analysis connects cumulative validation progress to target attainment and training-time cost. In this sense, BCLAN achieves a near-optimal efficiency-performance trade-off that is both analyzable and practically meaningful.

The experiments are strongly consistent with this mechanism. Across twelve heterogeneous medical benchmarks, BCLAN is the only compared method that consistently improves downstream performance while also reducing training time per epoch across unimodal and multimodal backbones. The gains are especially informative on regimes where allocation should be hardest, such as omics-based survival prediction, paired WSI-omics learning, and paired image-clinical or image-report settings, because these tasks exhibit pronounced heterogeneity in signal quality, difficulty, and convergence rate. At the same time, the improvements persist on already strong fusion backbones such as MAIL, HyPCA-Net, DRIFA-Net, and HEALNet, suggesting that BCLAN is not merely compensating for weak architectures. Instead, it improves the optimization layer on top of competitive multimodal models. The consistency across both paired and unpaired settings also argues against the gains being a dataset-specific artifact.

The ablation studies further explain why the hybrid design is necessary. Bandit-only variants improve efficiency more readily than predictive quality, which is consistent with coarse search rapidly locating promising regions but remaining exposed to high-variance updates. PID-only variants produce more stable gains but lack the global search needed to track a drifting optimum. The full controller performs best because it combines these two properties. The finer PID analysis is particularly revealing: the integral term reduces steady-state allocation bias, whereas the derivative term lowers trajectory variation and template switching. Likewise, the template-pool study exhibits the same finite-budget trade-off predicted by the theory: a very small pool under-covers the useful allocation space, whereas an overly large pool increases exploration burden. The empirical optimum at an intermediate pool size is therefore not incidental; it is precisely what one would expect when balancing approximation quality against online search cost.

Additional analyses reinforce the stability interpretation. BCLAN remains strong across a broad sweep of reward scales, indicating that its gains are not brittle to the precise calibration of the outer-loop signal. The validation-interval study shows that maximal responsiveness is not necessarily optimal: overly frequent updates expose the controller to short-horizon noise, whereas infrequent updates provide stale feedback. The best trade-off occurs at an intermediate update cadence, which is exactly the behavior expected from a well-tuned feedback system rather than a heuristic scheduler. Moreover, the controller overhead remains small across the total training time, so the efficiency gains are not offset by the cost of adaptation. Finally, the loss-landscape visualizations show convergence to broader, flatter low-loss basins than those obtained by open-loop or bandit-only baselines. Although such visualizations should not be interpreted as definitive evidence of improved generalization, they are consistent with the claim that BCLAN induces smoother and more stable optimization dynamics.