LEARNING FROM GOOD PLACEMENTS: MEMORY AND ADAPTIVE SELF-IMITATION FOR MACRO PLACEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for macro placement relies on intermediate placement information to guide hundreds of sequential decisions, whereas downstream quality can only be assessed after completing the full sequence and running a costly global-placement stage that includes standard-cell placement. Moreover, downstream placement quality is not necessarily well correlated with intermediate macro placement objectives, and these relationships can vary across benchmark designs, while the relative importance of these objectives is typically fixed throughout policy optimization. This limits both alignment with the downstream objective and the ability to adapt to the characteristics of individual placement problems. We address this with COMPAS, a **C**onstrained **O**ptimization framework for **M**acro **P**lacement via **A**daptive **S**elf-Imitation, which incorporates sparse downstream placement feedback directly into policy learning. Downstream evaluation is incorporated into COMPAS through two pathways. Global Placement (GP) improving placements update a GP-selected reference memory that guides subsequent search, while selected evaluated trajectories directly supervise the policy through adaptive self-imitation (ASI). During standard policy optimization, PPO is trained using dense wirelength and reference-placement guidance rewards. Periodically, complete macro-placement trajectories are evaluated by the downstream global placer, and selected trajectories are retained for replay, allowing decisions associated with favorable downstream outcomes to directly shape subsequent policy updates. Experiments on eight ICCAD 2015 contest benchmarks show that COMPAS reduces mean GP-HPWL by approximately 8.7% relative to EXPlace, the strongest RL-based baseline in our evaluation, improving seven of eight benchmarks with reductions of up to 22%. Additionally, COMPAS discovers 44% more new GP-HPWL incumbents than the non-ASI variant, with the cumulative gap widening over the optimization horizon. This trend suggests that ASI continues to improve downstream search later in training by replaying trajectories associated with favorable global-placement outcomes. In downstream physical-quality evaluation, COMPAS further achieves 9.6% lower average rWL, 7.9% lower NVP, and a 49% reduction in TNS magnitude relative to EXPlace. Together, these results show that sparse downstream evaluation can both guide subsequent search and provide direct policy supervision through adaptive self-imitation during sequential macro-placement optimization.

1 INTRODUCTION

Macro placement is an influential early stage of physical design. The positions of a relatively small number of large macros constrain the space available to standard cells and can strongly affect the quality achieved by subsequent placement stages (Lu et al., 2015; Cheng et al., 2019; Lin et al., 2019). Macro placement is inherently combinatorial and highly non-local, since moving one macro changes the feasible and desirable locations of many others. This naturally motivates a sequential decision-making formulation, in which a policy constructs the placement of one macro at a time (Mirhoseini et al., 2021; Lai et al., 2022; Geng et al., 2024).

A central challenge is that the information available during sequential placement does not directly reflect the quantity ultimately used to evaluate the completed design. Intermediate quantities such as wirelength and legality can be evaluated at each decision and provide dense feedback for policy

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.

learning. Existing macro-placement methods often rely on intermediate surrogate quantities that can be evaluated efficiently during macro placement (Lai et al., 2022; Shi et al., 2023; Xue et al., 2024). The downstream effect of the resulting macro configuration becomes apparent only after subsequent placement stages, whose repeated evaluation can be prohibitively expensive (Geng et al., 2025; Gao et al., 2026). We use downstream global placement (GP) to evaluate the completed macro configuration, and use its half-perimeter wirelength (GP-HPWL) as the downstream metric. Obtaining GP-HPWL is substantially more expensive than evaluating the intermediate placement information available during macro placement, which makes downstream evaluation impractical at every action.

This creates a mismatch between the supervision available during policy optimization and the quality measured downstream (Geng et al., 2025; Gao et al., 2026). A policy may improve the dense training objective without achieving a corresponding improvement under global placement. At the same time, a placement with favorable GP-HPWL provides direct evidence about the sequence of macro decisions that led to a favorable downstream outcome, but this information is available only after the trajectory is complete. Thus, the learning problem includes both constructing informative stepwise rewards and incorporating sparse downstream evidence into policy learning. Recent methods address this cross-stage mismatch through learned cross-stage predictors for placement guidance (Geng et al., 2025) or timing-guided preference optimization (Gao et al., 2026).

We introduce COMPAS, **C**onstrained **O**ptimization for **M**acro **P**lacement via **A**daptive **S**elf-Imitation. COMPAS instead periodically evaluates the policy's completed placements through global placement and feeds this information back into ongoing sequential optimization in two ways. GP-improving placements update a GP-selected reference memory that guides subsequent search, while selected evaluated trajectories are incorporated into COMPAS's adaptive self-imitation (ASI) pathway through self-imitation learning (SIL) (Oh et al., 2018), alongside PPO's on-policy learning. During ordinary interaction, PPO (Schulman et al., 2017) learns from dense wirelength and reference-placement guidance rewards, while ASI provides additional policy supervision from downstream-evaluated experience. Scoring the replay trajectories using the dense PPO reward would retain the same mismatch with downstream placement quality. A fixed combination of GP-HPWL and macro placement (MP) HPWL (MP-HPWL) would instead assign constant importance to the macro-level proxy, even though its relationship with downstream quality can vary across benchmark designs and over the course of optimization. ASI uses GP-HPWL as its primary replay criterion while treating MP-HPWL as an adaptive macro-quality constraint. Evaluator-guided self-imitation additionally adapts the relative weighting and overall strength of the policy's placement priors. Thus, sparse downstream evaluation influences both which trajectories receive additional supervision and how strongly the policy relies on its explicit priors.

Our experiments on eight ICCAD 2015 contest benchmarks demonstrate that COMPAS reduces mean GP-HPWL by approximately 8.7% relative to EXPlace, with lower GP-HPWL on seven of eight benchmarks and reductions of up to 22%. Under the same downstream-evaluation budget, COMPAS discovers 44% more new GP-HPWL incumbents than its variant without ASI. The gap grows over the optimization horizon, which indicates that ASI supports continued improvement of the best observed downstream placement throughout optimization. Downstream physical-quality evaluation further yields 9.6% lower average rWL, 7.9% lower NVP, and a 49% reduction in TNS magnitude relative to EXPlace. Furthermore, ablations show performance degradation when GP-selected reference memory or ASI is removed, and when the adaptive macro-quality constraint is replaced by a fixed multiplier.

Our contributions are as follows:

- We introduce COMPAS, a downstream-aware framework that feeds periodic global-placement evaluations back into sequential macro placement through GP-selected reference memory and adaptive self-imitation. Selected references guide subsequent rollouts through our reference-placement prior (RPP) and the reference positions used in wirelength guidance.
- We propose adaptive self-imitation (ASI), which prioritizes GP-HPWL under an adaptive MP-HPWL constraint and uses selected downstream-evaluated trajectories to update both the shared policy representation and its placement-prior coefficients.

- We conduct controlled ablations and optimization-dynamics analyses. The experiments examine the contributions of GP-selected reference memory, RPP, ASI, and the adaptive macro-quality constraint, and further evaluate downstream physical quality, sensitivity and generalization across held-out designs and benchmark families.

2 Related Work

Existing RL-based macro placers use spatial state representations and dense placement guidance to support sequential macro decisions. MaskPlace introduced pixel-level position, wirelength, and view masks for sequential macro placement, with the wire mask estimating the placement-dependent wirelength cost of candidate locations (Lai et al., 2022). WireMask-BBO subsequently applied wire-mask guidance to black-box optimization without policy learning (Shi et al., 2023). MaskRegulate instead formulates RL as a refinement of an existing macro placement and introduces regularity information into both the state and reward (Xue et al., 2024). Their learning or search objectives primarily use placement-stage HPWL and geometric proxies. COMPAS extends this line of work with periodic global-placement feedback for subsequent search and policy learning.

Recent methods such as LaMPlace and EXPlace incorporate later-stage information into macro placement in different ways. LaMPlace learns a predictor of cross-stage metrics and uses the predicted quality to construct placement guidance (Geng et al., 2025). EXPlace incorporates established placement expertise into the RL state and reward and further emulates expert refinement through preference optimization based on timing feedback (Gao et al., 2026). COMPAS differs in how downstream evaluation is incorporated into sequential policy optimization. Periodic global-placement evaluations update GP-selected reference memory and supervise a separate self-imitation replay pool. Replay prioritizes GP-HPWL improvement with an adaptive penalty for excessive MP-HPWL degradation. The relative weighting and overall strength of wirelength and reference-placement priors are updated through evaluator-guided self-imitation.

Self-imitation learning reinforces an agent's past actions when the corresponding empirical returns exceed the agent's current value estimates (Oh et al., 2018), while prioritized experience replay assigns greater sampling probability to more informative transitions (Schaul et al., 2016). Constrained RL separately studies policy optimization under auxiliary cost or feasibility requirements (Achiam et al., 2017; Tessler et al., 2019; Stooke et al., 2020). COMPAS differs from these formulations by using sparse downstream GP evaluation to determine which completed macro-placement trajectories receive additional supervision, while an adaptive MP-HPWL constraint regulates their replay. Evaluator-guided self-imitation additionally controls the policy's explicit placement-prior coefficients.

3 Method

3.1 Problem Formulation and COMPAS Overview

We formulate macro placement as a sequential decision problem. For a design with T macros, the policy places one macro at each step $t \in \{1, \dots, T\}$. The state s_t contains the current placement information, and the action $a_t \in \mathcal{A}(s_t)$ selects a grid location for the macro being placed. The feasible action set $\mathcal{A}(s_t)$ is determined by the position mask, which excludes locations that would overlap already placed macros or violate the placement boundary. A complete episode produces a trajectory $\tau = ((s_t, a_t))_{t=1}^{T}$.

We distinguish two metrics, both of which are minimized. MP-HPWL, $M(\tau)$, measures macro-placement HPWL over macro pins before downstream placement. GP-HPWL, $G(\tau)$, denotes the HPWL returned by the downstream placement evaluation pipeline, including global placement and legalization. Our downstream objective is to find macro placements with low GP-HPWL. The metric definitions and evaluator settings are detailed in Appendix A. COMPAS combines dense on-policy learning with periodic downstream evaluation, as summarized in Fig. 1. During ordinary interaction, PPO (Schulman et al., 2017) learns from a dense reward combining legality, wirelength, and our reference-placement prior (RPP) guidance:

$$r_t = r_t^{\text{legal}} + \omega r_t^{\text{wire}} + (1 - \omega) r_t^{\text{ref}}, \tag{1}$$

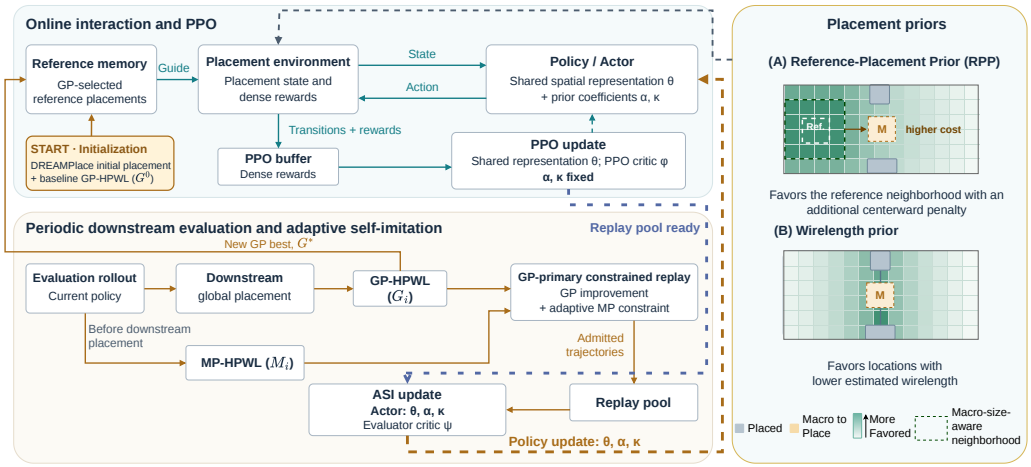


Figure 1: Overview of COMPAS. Each rollout uses a reference placement to construct the reference-placement prior (RPP) and wirelength guidance, while PPO learns from dense rewards. Periodic downstream evaluation provides MP-HPWL and GP-HPWL feedback through two pathways: GP-improving placements update reference memory, and selected trajectories enter the ASI replay pool. ASI updates the shared policy representation and placement-prior coefficients. The right panel illustrates the wirelength prior and our reference-placement prior (RPP), which favors a neighborhood around each macro's reference position.

where ω is fixed during training. The legality term is determined by the position mask, taking value zero for feasible actions and a large negative value otherwise; the same mask is applied to the policy before action selection to exclude infeasible locations. Reward construction, masking, and normalization are specified in Appendix A.

3.2 Policy with Placement Priors

The actor assigns a learned score to each grid location and adjusts it using wirelength and reference-placement priors. Let $z_\theta(s_t, a)$ denote the logit produced for location a by the spatial network with parameters θ. The priors are represented by cost fields $W_t(a)$ and $R_t(a)$, respectively, with lower values indicating preferred locations. The final action logit is

$$\ell_{\theta,\alpha,\kappa}(s_t, a) = z_\theta(s_t, a) - \kappa \left[\alpha W_t(a) + (1-\alpha) R_t(a)\right]. \tag{2}$$

The policy $\pi_{\theta,\alpha,\kappa}$ applies a softmax over feasible locations; infeasible actions are masked. The bounded coefficients α and κ control the relative weighting and overall strength of the priors. They are distinct from the fixed reward coefficient ω in Eq. 1.

Reference memory. Reference memory starts from the initial placement and is updated whenever an evaluated placement achieves lower GP-HPWL than the best stored reference. At the start of each rollout, a reference is selected with preference for lower GP-HPWL. The selected reference determines the reference-placement prior and supplies the positions of unplaced macros used to construct the wirelength field. Memory capacity and reference sampling are detailed in Appendix A.

Wirelength prior. The wirelength prior W_t is obtained from GeoWireMask, a topology-aware spatial field constructed from the current macro's incident nets. Its construction is detailed in Appendix C.

Reference-placement prior (RPP). For the current macro, RPP assigns a spatial cost $C_t(a)$ to each candidate location. Locations outside a macro-size-aware neighborhood of the reference position incur a penalty, with an additional penalty for centerward displacement.

The normalized field $R_t = \mathcal{N}(C_t)$ is supplied to the spatial network and implements the explicit reference-placement prior in Eq. 2. The cost $C_t(a_t)$ also determines the reference-placement guidance reward. This prior provides soft guidance; placement feasibility is enforced by a separate

action mask. The cost construction and reference sampling procedure are detailed in Appendix A.5; spatial-field normalization is specified in Appendix A.4.

3.3 Adaptive Self-Imitation

Self-imitation learning reinforces an agent's past actions when their empirical returns exceed its current value estimates (Oh et al., 2018). COMPAS uses downstream-evaluated experience to provide supervision beyond the dense PPO reward. Adaptive self-imitation (ASI) combines GP-primary constrained replay with evaluator-guided adaptation of the placement-prior coefficients.

3.3.1 GP-Primary Constrained Replay

ASI favors trajectories with low GP-HPWL while penalizing MP-HPWL degradation beyond an allowed tolerance. For an evaluated trajectory τ_i, let G_i and M_i denote its GP-HPWL and MP-HPWL. The initial placement GP-HPWL, G_0, provides a fixed baseline for quantifying downstream improvement, while the MP-HPWL anchor, A, provides the reference for measuring macro-placement degradation. The relative GP-HPWL improvement and MP-HPWL degradation are:

$$\Delta_i^G = \frac{G_0 - G_i}{G_0}, \qquad d_i^M(A) = \frac{M_i - A}{A}. \tag{3}$$

The replay score is

$$S_i(A, \lambda) = \Delta_i^G - \lambda[d_i^M(A) - \delta]_+, \tag{4}$$

where $[x]_+ = \max(0, x)$, $\delta \geq 0$ is the allowed relative macro degradation, and $\lambda \geq 0$ is the adaptive constraint multiplier. Higher scores are preferred. GP-HPWL supplies the improvement term, while MP-HPWL contributes a penalty only when degradation exceeds the tolerance. Thus, the macro-quality constraint is soft.

Adaptive constraint multiplier update. Let $G^\star$ denote the lowest GP-HPWL observed by ASI before the current evaluation. After initialization, only trajectories satisfying

$$G_i \leq (1 + \epsilon_G) G^\star \tag{5}$$

contribute to multiplier updates, where $\epsilon_G \geq 0$ is a relative tolerance. The first valid evaluated trajectory initializes $G^\star = G_i$, $A = M_i$, $\lambda = \lambda_0$, and $\bar{v} = 0$, without a multiplier update. For each subsequent competitive evaluated trajectory, we compute the signed constraint residual $v_i = d_i^M(A) - \delta$ before tightening the anchor. Applying a deadband threshold ϵ_v gives $\widetilde{v}_i = v_i \mathbf{1}\{|v_i| > \epsilon_v\}$. The updates are defined as

$$\bar{v} \leftarrow \rho \bar{v} + (1 - \rho)\widetilde{v}_i, \tag{6}$$

$$\lambda \leftarrow \mathrm{clip}\left(\lambda + \eta_\lambda \bar{v}, 0, \lambda_{\max}\right). \tag{7}$$

Here, $\bar{v}$ is an exponential moving average of the signed constraint residuals, ρ controls smoothing, η_λ is the update step size, and $\lambda_{\max}$ bounds the multiplier. Positive smoothed residuals strengthen the MP-HPWL penalty and negative residuals weaken it.

Anchor update and replay admission. When an evaluated trajectory establishes a new GP-HPWL incumbent, COMPAS updates $G^\star \leftarrow G_i$ and $A \leftarrow \min(A, M_i)$. The retained trajectories are subsequently rescored using the current values of A and λ. An evaluated trajectory is eligible for replay admission if the replay pool $\mathcal{D}$ is empty or its score is sufficiently close to the best score currently in the pool:

$$S_i(A, \lambda) \geq \max_{\tau_j \in \mathcal{D}} S_j(A, \lambda) - \epsilon_S, \tag{8}$$

where $\epsilon_S \geq 0$ is an absolute score margin. Eligible trajectories are retained subject to the replay capacity and replacement rule described in Appendix B.

3.3.2 Evaluator-Guided Self-Imitation Objective

ASI converts the current constrained scores into trajectory-level targets. Let $\mu_\mathcal{D}$ and $\sigma_\mathcal{D}$ denote the mean and standard deviation of scores in the replay pool. The standardized score and evaluator target are

$$\widehat{S}_i = \frac{S_i(A, \lambda) - \mu_\mathcal{D}}{\sigma_\mathcal{D} + \epsilon_\sigma}, \qquad y_i = c_{\mathrm{SIL}} \widehat{S}_i. \tag{9}$$

Here, $\epsilon_\sigma > 0$ stabilizes normalization, and $c_{\mathrm{SIL}} > 0$ sets the target scale. The trajectory-level target y_i is assigned to every stored decision in trajectory i. We set $\widehat{S}_i = 0$ when the pool contains fewer than two trajectories or $\sigma_{\mathcal{D}} < \epsilon_\sigma$. Targets are recomputed when retained scores or pool membership change. A separate evaluator critic V_ψ predicts these targets, while the PPO critic V_ϕ learns from dense rewards. Here, ϕ and ψ denote the parameters of the PPO critic and evaluator critic, respectively. ASI reinforces stored actions whose targets exceed the evaluator critic's predictions, using a capped positive advantage:

$$\widehat{A}^{\mathrm{SIL}}_{i,t} = \min\left\{[y_i - V_\psi(s_{i,t})]_+, a_{\max}\right\}, \tag{10}$$

where $a_{\max} > 0$ caps the advantage. The actor increases the likelihood of stored actions with positive evaluator advantages, while the evaluator critic learns to predict their targets. For a minibatch $\mathcal{M}$ of stored state–action pairs sampled with replacement, the losses are:

$$\mathcal{L}^{\mathrm{SIL}}_{\mathrm{actor}} = -\frac{1}{|\mathcal{M}|} \sum_{(i,t)\in\mathcal{M}} w_{i,t}\ \mathrm{sg}\left(\widehat{A}^{\mathrm{SIL}}_{i,t}\right) \log \pi_{\theta,\alpha,\kappa}(a_{i,t} \mid s_{i,t}), \tag{11}$$

$$\mathcal{L}^{\mathrm{SIL}}_{\mathrm{value}} = \frac{1}{|\mathcal{M}|} \sum_{(i,t)\in\mathcal{M}} w_{i,t}\ \mathrm{smoothL1}\left(V_\psi(s_{i,t}) - y_i\right). \tag{12}$$

Here, $w_{i,t}$ compensates for non-uniform replay sampling under prioritized experience replay (Schaul et al., 2016) and sg holds the advantage fixed during the actor update. Stored decisions whose evaluator targets exceed the critic's predictions by a larger amount are sampled more often for self-imitation. The evaluator critic is trained on errors in both directions, allowing its predictions to follow changes in the replay targets. Sampling and weight normalization are detailed in Appendix B.

3.3.3 Evaluator-Guided Placement-Prior Adaptation

PPO and evaluator-guided self-imitation both update the shared spatial representation. The placement-prior coefficients α and κ remain fixed during PPO and are updated only through Eq. 11. Thus, downstream-evaluated experience adapts the relative weighting and overall strength of the wirelength and reference-placement priors. The evaluator critic is updated separately using Eq. 12.

3.4 Training Procedure

Training alternates between trajectory collection and policy updates. PPO updates are performed when the rollout buffer is ready, followed by ASI updates when the replay pool contains sufficient experience. Periodic downstream evaluations update GP-selected reference memory and determine replay admission. The complete procedure is given in Appendix A.1.

4 Experiments

4.1 Experimental Setup

We primarily evaluate on the ICCAD 2015 contest benchmarks (Kim et al., 2015), with cross-benchmark generalization to ISPD 2005 (Nam et al., 2005) reported in Appendix D.4. Initial reference placements are generated with DREAMPlace following MaskRegulate (Xue et al., 2024), and downstream evaluation follows the same DREAMPlace global-placement flow used in prior work (Xue et al., 2024; Gao et al., 2026). We evaluate GP-HPWL, our primary downstream metric, every five training episodes, using the same downstream-evaluation schedule across COMPAS and its ablations. Experiments run for 300 episodes over three random seeds. Paired comparisons use the same initial reference placement and experimental configuration. Hyperparameters are shared across experiments unless explicitly stated otherwise and are reported in Appendix A.2.

4.2 Downstream-Quality Benchmarking

We first evaluate downstream placement quality on the eight ICCAD 2015 contest benchmarks. We compare against EXPlace (Gao et al., 2026), the strongest recent RL baseline in our evaluation, together with MaskPlace and MaskRegulate. We use originally reported results where available;

otherwise, we use the corresponding results reported by EXPlace to keep the comparison aligned with published evaluations. Table 1 reports GP-HPWL, our primary downstream metric. COMPAS achieves the lowest average GP-HPWL, reducing the EXPlace average from 5.98×10^8 to 5.46×10^8, an improvement of approximately 8.7%. It achieves lower GP-HPWL on seven of the eight benchmarks, with reductions of up to 21.9% on Superblue5. For reference, we report DREAMPlace (Lin et al., 2019) GP-HPWL from our corresponding initializations, since ICCAD 2015 results are not reported in DREAMPlace.

Table 2 reports the mean of each downstream physical-quality metric across the eight ICCAD 2015 benchmark designs. We measure routed wirelength (rWL), horizontal and vertical routing overflow (rO-H and rO-V), and the number of violating paths (NVP) with Cadence Innovus. We evaluate worst negative slack (WNS) and total negative slack (TNS) with OpenTimer (Huang & Wong, 2015), following the evaluation flow of EXPlace. COMPAS achieves the best average rWL, NVP, WNS, and TNS among the compared methods. Relative to EXPlace, it reduces average rWL by 9.6%, NVP by 7.9%, and TNS magnitude by 49%, while routing overflow remains low. The training trajectories are shown in Fig. 2. Full per-benchmark physical-quality results and GP-HPWL mean $\pm$ standard deviation across runs are reported in Appendix D.1.

Table 1: Benchmarking of GP-HPWL ($\times10^8$) on the ICCAD 2015 contest benchmark suite. Superblue benchmarks are abbreviated as SB. Avg. denotes the mean across benchmarks.

Method	SB1	SB3	SB4	SB5	SB7	SB10	SB16	SB18	Avg.
DREAMPlace	10.83	12.53	6.68	8.71	20.91	13.03	8.22	3.78	10.59
MaskPlace	7.34	8.23	5.06	7.85	9.81	10.07	5.32	3.28	7.12
MaskRegulate	5.77	7.05	4.15	6.94	<u>7.90</u>	11.23	5.53	2.96	6.44
EXPlace	<u>5.34</u>	<u>6.54</u>	<u>3.82</u>	<u>6.90</u>	8.33	<u>9.69</u>	**4.49**	<u>2.70</u>	<u>5.98</u>
COMPAS	**5.13**	**5.89**	**3.67**	**5.39**	**6.74**	**9.45**	4.90	**2.50**	**5.46**

Table 2: Average downstream physical quality over eight ICCAD 2015 benchmarks. Lower is better for rWL, rO-H, rO-V, and NVP; WNS and TNS are better when closer to zero. Best and second-best results are shown in bold and underlined, respectively.

Approach	rWL (m)	rO-H (%)	rO-V (%)	NVP ($\times10^3$)	WNS (ns)	TNS ($\times10^4$ ns)
DREAMPlace	164.79	10.46	4.40	25.29	-99.01	-22.39
MaskRegulate	133.85	**0.24**	<u>0.06</u>	13.25	-75.56	-14.46
EXPlace	<u>121.19</u>	<u>0.36</u>	**0.05**	<u>11.83</u>	<u>-55.76</u>	<u>-8.59</u>
COMPAS	**109.55**	0.61	0.10	**10.90**	**-52.56**	**-4.36**

4.3 Optimization Dynamics of Adaptive Self-Imitation

Fig. 2 shows the evolution of best-so-far GP-HPWL during training. COMPAS obtains lower final GP-HPWL than EXPlace on seven of eight benchmark designs. GP-HPWL is observed only through periodic downstream evaluation rather than optimized as a dense reward, so these gains reflect the continued discovery of macro placements that remain favorable after global placement. The comparison with COMPAS w/o ASI isolates the contribution of adaptive self-imitation under the same placement-prior structure and downstream-evaluation schedule. At the end of training, full COMPAS achieves lower best-so-far GP-HPWL than its variant without ASI on six of eight designs. To characterize this advantage beyond the final best solution, we examine GP-incumbent discovery, its accumulation over the optimization horizon, and best-so-far macro-placement quality.

Fig. 3a reports the mean number of new GP-HPWL incumbents per run for each benchmark design. COMPAS discovers more incumbents on six of eight designs, with one tie and one design favoring the variant without ASI. Fig. 3b shows how this difference accumulates over training. Across 24 runs per method, COMPAS discovers 75 new incumbents versus 52 without ASI (3.13 vs. 2.17 per run). After within-design aggregation, the paired difference is $+0.96$ incumbents per run (95% design-cluster bootstrap CI: $[0.25,\ 1.67]$). The gap increases from $+0.13$ at episode 50 to $+0.96$ at episode 300. Thus, ASI supports continued downstream incumbent discovery throughout optimization. Fig. 3c reports best-so-far MP-HPWL improvement relative to each run's initial DREAMPlace placement. The mean MP-HPWL trajectory also favors COMPAS over most of the optimization horizon and continues to improve at later episodes. The uncertainty across benchmarks is reported in Appendix Fig. 6a.

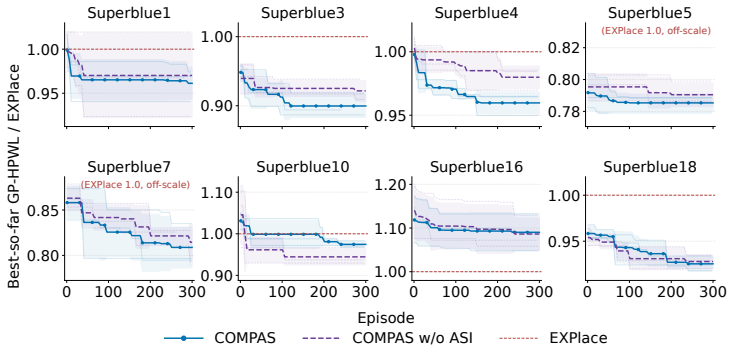
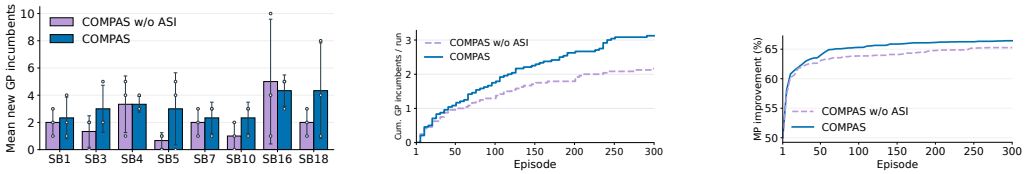


Figure 2: Best-so-far GP-HPWL during training, normalized by the EXPlace baseline. Lower is better. Lines show the mean over three seeds, and shading denotes ± 1 standard deviation. The EXPlace reference line is omitted where it falls outside the displayed range.



(a) Mean new GP incumbents (b) Cumulative GP inc. (c) Best-so-far MP

Figure 3: Effect of adaptive self-imitation on optimization dynamics. (a) Mean new GP-HPWL incumbents per run for each benchmark; error bars denote ± 1 standard deviation over three runs. (b) Mean cumulative GP-HPWL incumbents per run across eight benchmarks. (c) Best-so-far MP-HPWL improvement relative to the initial placement across the eight benchmarks.

4.4 Ablation of Downstream-Feedback Components

In Fig. 4 COMPAS achieves the lowest final GP-HPWL across all three ablation benchmark designs. Ablation of either ASI or GP-selected reference memory degrades downstream quality, and use of a fixed λ instead of the adaptive multiplier also yields worse GP-HPWL. COMPAS additionally achieves higher mean best-so-far MP-HPWL improvement than the fixed-λ variant across all eight designs (Appendix Fig. 6). Furthermore, Appendix D.3 shows that the multiplier adapts differently across designs and over the course of optimization.

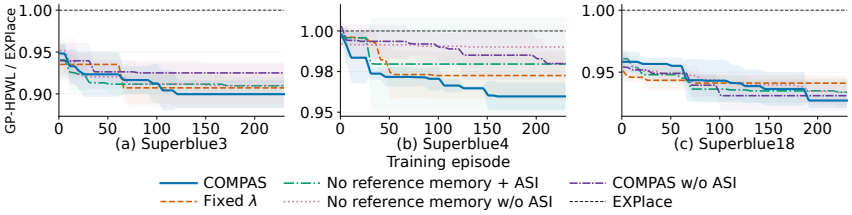


Figure 4: Ablation of COMPAS downstream-feedback components. Mean best-so-far GP-HPWL over three runs, normalized by EXPlace. Lower is better; shading denotes ± 1 standard deviation.

We also conduct an ablation study of the placement priors, which focuses on the proposed reference-placement prior (RPP). As shown in Appendix Fig. 9, COMPAS achieves lower best-so-far GP-HPWL than RegularMask on six of eight benchmarks, with a 4.52% average reduction and up to 11.11% improvement. Overall, the results indicate that downstream-evaluated reference placements provide more effective spatial guidance than generic boundary-regularity guidance. The same figure also reports the wirelength prior ablation.

4.5 Sensitivity to ASI Hyperparameters

We evaluate sensitivity to two ASI hyperparameters: the relative macro-degradation tolerance δ and the replay-admission margin ϵ_S. Fig. 5 reports best-so-far GP-HPWL on Superblue3 and Superblue18 while varying one parameter at a time around the default setting. For $\delta \in \{0, 0.05, 0.1\}$, the default $\delta = 0.05$ gives the lowest final GP-HPWL on Superblue18, while $\delta = 0.1$ performs best on Superblue3. For $\epsilon_S \in \{0.005, 0.015, 0.03\}$, the default $\epsilon_S = 0.015$ performs best on Superblue18, whereas $\epsilon_S = 0.005$ gives the lowest final value on Superblue3. All tested settings outperform the EXPlace reference on both designs. Overall, results indicate design-dependent sensitivity without reliance on a single parameter setting.

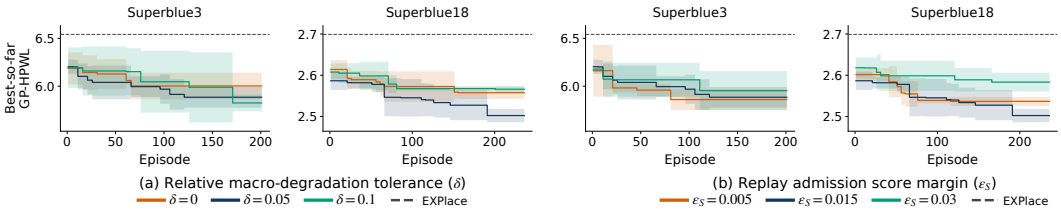


Figure 5: Sensitivity to ASI hyperparameters on Superblue3 and Superblue18. Best-so-far GP-HPWL ($\times 10^8$) under variations of (a) the relative macro-degradation tolerance δ and (b) the replay admission score margin ϵ_S.

4.6 Computational Cost

We compare the runtime and memory cost of COMPAS, MaskRegulate, and EXPlace on the eight ICCAD 2015 Superblue benchmarks under the same CPU environment. We report wall-clock time for two episodes. Timing includes initialization, preprocessing, model construction, two training rollouts, and one scheduled policy/DREAMPlace evaluation after episode 1, as well as method-specific initialization evaluations. COMPAS requires 399.54 s and 13.05 GiB on average, corresponding to 9.0% lower runtime and 7.2% lower peak memory than MaskRegulate, and 40.2% lower runtime and 63.4% lower peak memory than EXPlace. For reference, COMPAS end-to-end training through 300 episodes takes 24.73 h on average across the eight benchmarks. Runtime and memory results are detailed in Appendix D.6.

The memory reduction partly reflects COMPAS's compact spatial state representation. COMPAS stores three spatial planes per transition, compared with eight for MaskRegulate and thirteen for the evaluated EXPlace configuration. At equal resolution and precision, this corresponds to 62.5% and 76.9% fewer spatial planes, respectively, which lowers rollout-buffer memory and the policy input footprint.

5 Conclusions

We introduced COMPAS, a downstream-aware framework that uses sparse global-placement feedback to better align sequential macro-placement learning with downstream quality. GP-selected reference memory guides subsequent search, while adaptive self-imitation reuses selected evaluated trajectories for policy supervision. ASI prioritizes GP-HPWL improvement under an adaptive MP-HPWL constraint and updates the policy's placement-prior coefficients from evaluator feedback. Across eight ICCAD 2015 benchmarks, COMPAS reduces mean GP-HPWL by approximately 8.7% relative to EXPlace, with reductions of up to 22%, while also improving several downstream routing and timing metrics. Under the same evaluation budget, COMPAS discovers 44% more new GP-HPWL incumbents than its variant without ASI. These results show that sparse downstream evaluation can both steer subsequent search and provide direct supervision to the policy during sequential macro-placement optimization.

Reproducibility statement

We provide the implementation and evaluation details required to reproduce COMPAS in the main paper and appendix. The appendix specifies the network architecture, optimization and replay configuration, hyperparameters, downstream evaluation procedure, and statistical aggregation used in the reported experiments.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Chung-Kuan Cheng, Andrew B. Kahng, Ilgweon Kang, and Lutong Wang. RePlAce: Advancing solution quality and routability validation in global placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(9):1717–1730, 2019. doi: 10.1109/TCAD.2018.2859220.

Chengrui Gao, Yunqi Shi, Ke Xue, Ruo-Tong Chen, Siyuan Xu, Mingxuan Yuan, Chao Qian, and Zhi-Hua Zhou. Expertise can be helpful for reinforcement learning-based macro placement. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Zijie Geng, Jie Wang, Ziyan Liu, Siyuan Xu, Zhentao Tang, Mingxuan Yuan, Jianye Hao, Yongdong Zhang, and Feng Wu. Reinforcement learning within tree search for fast macro placement. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15402–15417. PMLR, 2024.

Zijie Geng, Jie Wang, Ziyan Liu, Siyuan Xu, Zhentao Tang, Shixiong Kai, Mingxuan Yuan, Jianye Hao, and Feng Wu. LaMPlace: Learning to optimize cross-stage metrics in macro placement. In *International Conference on Learning Representations*, 2025.

Tsung-Wei Huang and Martin D. F. Wong. OpenTimer: A high-performance timing analysis tool. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 895–902, 2015. doi: 10.1109/ICCAD.2015.7372666.

Myung-Chul Kim, Jin Hu, Jiajia Li, and Natarajan Viswanathan. ICCAD-2015 CAD contest in incremental timing-driven placement and benchmark suite. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 921–926, 2015. doi: 10.1109/ICCAD.2015.7372671.

Yao Lai, Yao Mu, and Ping Luo. MaskPlace: Fast chip placement via reinforced visual representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24019–24030, 2022.

Yibo Lin, Shounak Dhar, Wuxi Li, Haoxing Ren, Brucek Khailany, and David Z. Pan. DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement. In *Proceedings of the 56th Annual Design Automation Conference*, pp. 1–6, 2019. doi: 10.1145/3316781.3317803.

Jingwei Lu, Pengwen Chen, Chin-Chih Chang, Lu Sha, Dennis Jen-Hsin Huang, Chin-Chi Teng, and Chung-Kuan Cheng. eplace: Electrostatics-based placement using fast fourier transform and nesterov's method. *ACM Transactions on Design Automation of Electronic Systems*, 20(2):1–34, 2015. doi: 10.1145/2699873.

Azalia Mirhoseini et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021. doi: 10.1038/s41586-021-03544-w.

Gi-Joon Nam, Charles J. Alpert, Paul Villarrubia, Bruce Winter, and Mehmet Can Yildiz. The ISPD2005 placement contest and benchmark suite. In *Proceedings of the International Symposium on Physical Design*, pp. 216–220, 2005. doi: 10.1145/1055137.1055182.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafull Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Yunqi Shi, Ke Xue, Lei Song, and Chao Qian. Macro placement by wire-mask-guided black-box optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 6825–6843, 2023. doi: 10.52202/075280-0299.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. *International Conference on Learning Representations*, 2019.

Ke Xue, Ruo-Tong Chen, Xi Lin, Yunqi Shi, Shixiong Kai, Siyuan Xu, and Chao Qian. Reinforcement learning policy as macro regulator rather than macro placer. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140565–140588, 2024. doi: 10.52202/079017-4462.

A Implementation Details

A.1 Algorithm

Algorithm 1 summarizes the complete COMPAS training procedure.

Algorithm 1 COMPAS

Require: Policy $\pi_{\theta,\alpha,\kappa}$; PPO critic V_ϕ; evaluator critic V_ψ
Initialize baseline G_0, reference memory $\mathcal{R}$, PPO buffer $\mathcal{B}$, replay pool $\mathcal{D}$, and ASI constraint state
for each training episode **do**
  Collect a trajectory using a reference from $\mathcal{R}$ and store transitions in $\mathcal{B}$
  if $\mathcal{B}$ is ready **then**
    Update θ, ϕ with PPO; hold α, κ fixed
    if $\mathcal{D}$ is ready **then**
      Update $\theta, \alpha, \kappa, \psi$ with ASI (Eqs. 11–12)
    end if
    Clear $\mathcal{B}$
  end if
  if downstream evaluation is scheduled **then**
    for each evaluation rollout **do**
      Generate τ_i and obtain MP-HPWL M_i and GP-HPWL G_i
      Update $\mathcal{R}$ if G_i improves its GP incumbent
      Initialize or update the ASI constraint state using Eqs. 6–7
      Rescore retained trajectories under the current constraint state and compute S_i using Eq. 4
      Update $\mathcal{D}$ for subsequent ASI updates using Eq. 8 and refresh replay targets
    end for
  end if
end for
return Best evaluated placement

A.2 Hyperparameters

Table 3 lists the hyperparameters used in COMPAS.

A.3 Placement Metrics and Downstream Evaluation

MP-HPWL. Following prior work (Xue et al., 2024), MP-HPWL is computed from the completed macro configuration represented in the placement environment. Macro-pin coordinates incorporate the macro location, dimensions, and pin offsets. For each represented net, HPWL is the sum of

Table 3: Key method-specific hyperparameters for COMPAS.

Group	Hyperparameter	Value
Constrained replay	Macro degradation tolerance δ	0.05
	Initial constraint multiplier λ_0	0.5
	Replay admission margin ϵ_S	0.015
	Multiplier step size η_λ	0.5
	Maximum multiplier $\lambda_{\max}$	5.0
	EMA coefficient ρ	0.9
	Deadband threshold ϵ_v	0.005
	GP-HPWL tolerance ϵ_G	0.01
Placement guidance	Dense-reward wirelength weighting ω	0.7
	Initial wirelength-prior weighting α_0	0.7
	Initial placement-prior strength κ_0	0.5
	GeoWireMask cancellation threshold χ_{th}	0.6
Replay and environment	ASI replay capacity K_{SIL}	20
	ASI minibatch updates per PPO update cycle	32
	Reference-memory capacity K_{ref}	10
	Evaluation interval	5 episodes
	Placement grid	224×224
	Maximum placement horizon	512 steps

the horizontal and vertical pin spans, with net weights applied when provided by the benchmark. Standard-cell pins absent from the macro-placement representation do not contribute to this metric.

Downstream evaluation. Following prior work (Xue et al., 2024), the completed macro configuration is exported to the DREAMPlace evaluation pipeline, which enables global placement and legalization. We use the final HPWL metric returned by this pipeline, GP-HPWL. G_0 is the GP-HPWL associated with the initial reference placement, established before training and held fixed throughout optimization. It is distinct from the evolving best evaluated GP-HPWL $G^\star$.

A.4 Dense Rewards and Spatial-Field Normalization

Let $\widetilde{W}_t$ denote the wirelength cost field before actor-input normalization. The unscaled wirelength and reference-placement rewards are $-\widetilde{W}_t(a_t)$ and $-C_t(a_t)$, respectively. Before policy training, one rollout collects the minimum and maximum values of each reward component for subsequent reward scaling. These bounds remain fixed, and subsequent rewards are not clipped. Separately, each spatial cost field is divided by its maximum absolute value before being supplied to the actor.

A.5 Reference-Placement Prior (RPP)

Reference memory. Reference memory starts from the initial placement. A new placement is inserted when it improves the reference GP-HPWL incumbent, and memory retains at most K_{ref} entries with the lowest GP-HPWL.

At reset, the current best reference is selected with probability $1 - p_{\text{ref}}$. Otherwise, a reference is sampled from the memory. For stored reference j with GP-HPWL H_j, the conditional probability is

$$P(j) \propto \exp\left[-\frac{H_j - H_{\min}}{\tau_{\text{ref}}(H_{\max} - H_{\min} + 10^{-9})}\right]. \tag{13}$$

Here, $H_{\min}$ and $H_{\max}$ are the minimum and maximum GP-HPWL values in the memory. The configuration uses $K_{\text{ref}} = 10$, $p_{\text{ref}} = 0.7$, and $\tau_{\text{ref}} = 0.5$.

Spatial cost field. The reference-placement prior (RPP) is implemented through $C_t(a)$, whose normalized form R_t is incorporated into Eq. 2. The same cost field determines the unscaled reference-placement guidance reward.

Let (x_0, y_0) be the macro's reference location and (b_x, b_y) its dimensions in grid units. For a placement grid of side length L, define $L_x = L - b_x$ and $L_y = L - b_y$. Reference and candidate coordinates are clipped to $[0, L_x] \times [0, L_y]$ when computing the cost. The macro-size-aware tolerances are

$$t_x = \max(1, \text{round}(mb_x)), \qquad t_y = \max(1, \text{round}(mb_y)),$$

where m is the margin fraction.

Let s_x and s_y denote the physical distances corresponding to one grid step along each axis. The cost uses coordinate weights q_x, q_y proportional to s_x, s_y, respectively:

$$q_x = \frac{s_x}{s_x + s_y}, \qquad q_y = \frac{s_y}{s_x + s_y}.$$

The reference-window cost is

$$C_{\text{window}} = q_x[|x - x_0| - t_x]_+ + q_y[|y - y_0| - t_y]_+.$$

Define the directions toward the center of the feasible coordinate range as

$$d_x = \begin{cases} 1, & x_0 \le L_x/2, \\ -1, & \text{otherwise}, \end{cases} \qquad d_y = \begin{cases} 1, & y_0 \le L_y/2, \\ -1, & \text{otherwise}. \end{cases}$$

The additional centerward-displacement cost is

$$C_{\text{center}} = q_x[d_x(x - x_0) - t_x]_+ + q_y[d_y(y - y_0) - t_y]_+.$$

The combined cost is

$$C_t(x, y) = \frac{C_{\text{window}} + c_{\text{center}} C_{\text{center}}}{(1 + c_{\text{center}})(q_x L_x + q_y L_y) + 10^{-9}}. \tag{14}$$

The configuration uses $m = 1$, $c_{\text{center}} = 1$.

A.6 Learned Spatial Representation and Prior Coefficients

The actor receives the reference-placement prior, position mask, and wirelength prior as three spatial input channels. A local convolutional branch scores individual grid locations, while a spatial encoder–decoder branch incorporates information from surrounding regions. The local branch uses 1×1 convolutions with channels $3, 12, 12, 1$ and intermediate ReLU activations. The spatial branch applies average pooling by four, followed by an encoder with channel widths $8, 16, 24, 32$ and a decoder with bilinear upsampling and skip connections. Each spatial block contains two 3×3 convolutions with ReLU activations.

The learned logits are

$$z_\theta = z_\theta^{\text{local}} + g_\theta \tanh(z_\theta^{\text{spatial}}), \qquad g_\theta = 0.25 \ \text{sigmoid}(u_\theta). \tag{15}$$

The gate is initialized to 0.05, and the spatial output head is initialized to zero. Both branches and the gate are part of the shared spatial representation.

Each prior coefficient $c \in \{\alpha, \kappa\}$ is parameterized by a trainable scalar d_c, mapped to the prescribed coefficient bounds as follows:

$$c(d_c) = c_0 + \begin{cases} (c_{\max} - c_0) \tanh(d_c), & d_c \ge 0, \\ (c_0 - c_{\min}) \tanh(d_c), & d_c < 0. \end{cases} \tag{16}$$

The scalar d_c is initialized to zero, updated only through the ASI actor loss, and held fixed during PPO. We use $\alpha_0 = 0.7$, $\alpha_{\min} = 0.05$, $\alpha_{\max} = 0.95$, $\kappa_0 = 0.5$, $\kappa_{\min} = 0$, and $\kappa_{\max} = 1$.

A.7 Optimization and Evaluation Settings

The actor, PPO critic, and evaluator critic use separate Adam optimizers with learning rate 2×10^{-4} and gradient-norm clipping at 0.5. PPO and self-imitation share the actor optimizer. PPO uses 5,120-transition rollout buffers, up to four epochs, minibatches of 256, probability-ratio clipping at 0.1, and early stopping when the estimated minibatch KL exceeds 0.02. GAE uses $\gamma = 0.99$ and $\lambda_{\text{GAE}} = 0.95$. Advantages are normalized over the rollout buffer unless their standard deviation is below 10^{-6}. The entropy coefficient decreases linearly from 10^{-3} to 3×10^{-4} over the episodes and remains constant thereafter. The PPO critic uses smooth-L_1 regression with threshold one and unit loss coefficient.

Downstream evaluation uses one greedy rollout after episode one and every five episodes thereafter. Each evaluated trajectory is considered for replay admission independently of reference-memory updates.

B Adaptive Self-Imitation Replay Details

Replay targets follow Eq. 9. The replay capacity and update count are listed in Table 3. Eligible trajectories are inserted when the replay pool contains fewer than K_{SIL} trajectories. Otherwise, an eligible candidate replaces the lowest-scoring retained trajectory only if its score is higher. Rescoring updates retained scores and their ordering without reapplying the admission criterion to existing trajectories. Replay follows each complete PPO update cycle once the replay pool contains at least 64 stored state–action pairs, with no additional episode-based warm-up. Replay minibatches contain 64 transitions. We use $c_{\mathrm{SIL}} = 20$, $a_{\max} = 5$, $\nu = 1$, fixed $\beta = 1$, $\epsilon_p = 10^{-3}$, $c_\pi = 0.2$, and $c_V = 0.02$. Pool-score normalization uses the population standard deviation and $\epsilon_\sigma = 10^{-4}$, following the stabilization and low-variance rules in Eq. 9. The evaluator critic uses importance-weighted smooth-L_1 regression with threshold one, penalizing both overprediction and underprediction.

Evaluator critic. A separate network receives the reference-placement prior, position mask, and wirelength prior, average-pooled to $3 \times 56 \times 56$. Three ReLU convolutional layers have channel widths $(16, 32, 32)$, kernel sizes $(5, 5, 3)$, strides $(2, 2, 2)$, and padding $(2, 2, 1)$. Adaptive average pooling to $32 \times 4 \times 4$ precedes a $512, 64, 1$ fully connected head with a ReLU hidden activation and linear output.

Prioritized sampling and losses. Let

$$\mathcal{I}_{\mathcal{D}} = \{(i,t) : \tau_i \in \mathcal{D},\ 1 \le t \le T\}$$

index retained transitions, and let $N_{\mathcal{D}} = |\mathcal{I}_{\mathcal{D}}|$. Prioritized replay (Schaul et al., 2016) uses

$$p_{i,t} = \frac{(\widehat{A}^{\mathrm{SIL}}_{i,t} + \epsilon_p)^\nu}{\sum_{(j,u)\in\mathcal{I}_{\mathcal{D}}} (\widehat{A}^{\mathrm{SIL}}_{j,u} + \epsilon_p)^\nu}, \tag{17}$$

where $\widehat{A}^{\mathrm{SIL}}_{i,t}$ is defined in Eq. 10. Priorities are recomputed using the current evaluator critic before each update, and minibatches are sampled with replacement.

The importance-sampling weights are

$$\widetilde{w}_{i,t} = (N_{\mathcal{D}} p_{i,t})^{-\beta}, \qquad w_{i,t} = \frac{\widetilde{w}_{i,t}}{\max_{(j,u)\in\mathcal{M}} \widetilde{w}_{j,u} + 10^{-8}}. \tag{18}$$

Normalization uses the maximum weight in the sampled minibatch. These weights compensate for nonuniform replay sampling.

Each replay step minimizes $c_\pi \mathcal{L}^{\mathrm{SIL}}_{\mathrm{actor}}$ with respect to θ, α, κ, followed by $c_V \mathcal{L}^{\mathrm{SIL}}_{\mathrm{value}}$ with respect to ψ. Targets, advantages, and sampling weights are held fixed during the actor update.

C Wirelength Prior

COMPAS uses GeoWireMask, a topology-aware spatial wirelength prior. For the macro being placed, the field is constructed from its incident nets. Already placed macros contribute their current positions, whereas unplaced macros use their positions in the selected reference placement. Lower field values indicate preferred candidate locations. The resulting field supplies the wirelength reward and, after normalization, is provided to the actor as the explicit wirelength prior in Eq. 2. Placement feasibility is enforced separately by the position mask.

Per-net geometry and potentials. For each incident net e, the bounding box of the other represented pins is translated by the current macro's pin offset and clipped to the grid, producing a candidate-location box B_e with center $\mathbf{c}_e$.

The resultant direction is computed from neighboring macro locations, obtained by subtracting their pin displacements:

$$\mathbf{r}_e = \sum_j (\mathbf{a}_{j,e} - \mathbf{c}_e).$$

A net is nondirectional when fewer than two other pins are represented, their bounding box has zero span along either axis, or $\|\mathbf{r}_e\|_2 < 10^{-3}$.

Nondirectional nets use coordinate-weighted Manhattan distance to B_e, which is zero inside the box. For directional nets, let $D_e(\mathbf{a})$ be the coordinate-scaled Euclidean distance from $\mathbf{c}_e$, normalized by the maximum distance to a grid corner. Their potential is

$$f_e(\mathbf{a}) = \begin{cases} D_e(\mathbf{a}), & \mathbf{a} \notin B_e, \\ -(1 - D_e(\mathbf{a})), & \mathbf{a} \in B_e,\ \mathbf{d}_e(\mathbf{a})^\top \mathbf{r}_e < 0, \\ -\frac{1}{2}(1 - D_e(\mathbf{a})), & \text{otherwise}, \end{cases} \tag{19}$$

where $\mathbf{d}_e(\mathbf{a})$ is the coordinate-scaled displacement from the box center.

Normalization and aggregation. Let $\mathcal{Q}$ denote spatial min–max normalization with stabilizer 10^{-5}. Each net potential is normalized and its value at the reference macro location is subtracted. The resulting potentials are summed separately over directional and nondirectional nets:

$$F = \mathcal{Q}(\zeta F_{\mathrm{d}} + (1 - \zeta) F_{\mathrm{n}}), \qquad \zeta = \frac{n_{\mathrm{d}}}{n_{\mathrm{d}} + n_{\mathrm{n}}}, \tag{20}$$

where n_{d} and n_{n} are the contributing net counts. This geometric mixture is distinct from the learned placement-prior coefficient α.

Directional cancellation and edge guidance. For unit resultant directions $\mathbf{u}_e$, cancellation is

$$\chi = 1 - \frac{\|\sum_e \mathbf{u}_e\|_2}{n_{\mathrm{d}} + 10^{-9}}, \tag{21}$$

with $\chi = 0$ when fewer than two directional nets exist. If $\chi \geq 0.6$, an edge-oriented field replaces F.

When directional cancellation is high, an edge-oriented field replaces the aggregated net potentials, favoring locations within the enclosing net bounding box toward the nearest chip boundary.

D Additional Results

D.1 Downstream quality comparison on the ICCAD 2015 contest benchmarks

Table 4 reports COMPAS GP-HPWL across the three runs for each ICCAD 2015 benchmark, while Table 5 reports the downstream physical-quality metrics of COMPAS on each benchmark.

Table 4: COMPAS GP-HPWL across three runs on the ICCAD 2015 contest benchmarks, reported as mean ± standard deviation ($\times 10^8$). Superblue benchmarks are abbreviated as SB.

Metric	SB1	SB3	SB4	SB5	SB7	SB10	SB16	SB18
GP-HPWL	5.133 ± 0.097	5.885 ± 0.108	3.667 ± 0.033	5.389 ± 0.043	6.736 ± 0.189	9.451 ± 0.073	4.896 ± 0.189	2.498 ± 0.022

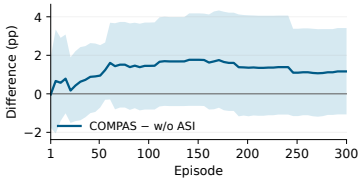
D.2 Macro-stage quality over training

Fig. 6a quantifies the uncertainty associated with Fig. 3c. At each episode, paired runs are averaged within each benchmark design, and the resulting eight design-level differences are bootstrapped with 20,000 resamples.

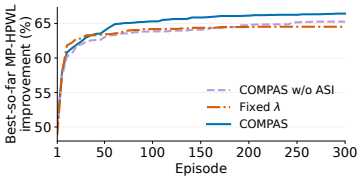
Table 6 summarizes the corresponding 300-episode-horizon results. Improvements are computed relative to each run's initial reference placement MP-HPWL and averaged first within design and then across benchmark designs. COMPAS attains the largest mean MP-HPWL improvement, exceeding the variant without ASI and the fixed-λ variant by 1.17 and 1.90 percentage points, respectively. Together with Fig. 6b, this complements the GP-HPWL ablations by showing that the downstream gains are accompanied by improved macro-stage quality.

Table 5: Downstream physical quality on the ICCAD 2015 benchmarks. Units: rWL (m), rO-H/rO-V (%), NVP ($\times 10^3$), WNS (ns), and TNS ($\times 10^4$ ns). Lower is better except WNS and TNS, which are better closer to zero. Best and second-best values are bold and underlined; Avg. denotes the mean across benchmarks.

Metric	Approach	SB1	SB3	SB4	SB5	SB7	SB10	SB16	SB18	Avg.
rWL	DREAMPlace	159.00	199.00	140.00	167.00	220.00	244.00	136.00	53.30	164.79
	MaskRegulate	121.00	149.00	87.30	142.00	186.00	225.00	102.00	58.50	133.85
	EXPlace	108.00	134.00	77.50	138.00	171.00	195.00	**90.20**	55.80	121.19
	COMPAS	**106.06**	**118.37**	**73.95**	**104.66**	**137.77**	**189.29**	94.87	**51.43**	**109.55**
rO-H	DREAMPlace	16.93	15.05	18.74	5.67	6.93	3.83	16.37	**0.19**	10.46
	MaskRegulate	**0.07**	**0.37**	0.39	0.11	0.20	0.18	**0.36**	0.20	**0.24**
	EXPlace	0.15	1.24	**0.27**	0.30	**0.01**	**0.13**	**0.36**	0.41	0.36
	COMPAS	1.34	1.52	**0.27**	**0.01**	0.19	0.32	0.83	0.43	0.61
rO-V	DREAMPlace	4.72	7.40	11.60	2.18	5.24	2.10	1.93	0.02	4.40
	MaskRegulate	**0.02**	**0.05**	0.07	0.12	0.03	**0.09**	**0.06**	**0.01**	0.06
	EXPlace	**0.02**	0.10	**0.05**	0.02	**0.01**	0.12	0.07	0.02	**0.05**
	COMPAS	0.17	0.18	0.07	**0.01**	0.08	0.15	0.08	0.09	0.10
NVP	DREAMPlace	30.00	19.80	26.90	20.50	38.10	24.50	37.50	5.02	25.29
	MaskRegulate	12.50	12.00	9.94	13.20	19.30	16.30	17.40	5.33	13.25
	EXPlace	13.20	9.86	8.77	12.40	**13.90**	15.30	**16.50**	4.67	11.83
	COMPAS	**11.48**	**6.84**	**8.37**	**7.41**	16.21	**13.98**	18.38	**4.49**	**10.90**
WNS	DREAMPlace	-111.00	-108.00	-106.00	-71.70	-59.60	-240.00	-65.50	-30.30	-99.01
	MaskRegulate	-66.40	-135.00	-63.00	-127.00	-70.40	**-67.30**	-42.90	-32.50	-75.56
	EXPlace	-58.60	**-82.30**	-34.10	-111.00	-41.80	-71.60	**-21.40**	-25.30	-55.76
	COMPAS	**-37.20**	-93.80	**-24.04**	**-56.37**	**-39.78**	-85.64	-68.76	**-14.91**	**-52.56**
TNS	DREAMPlace	-37.30	-25.10	-21.80	-17.40	-13.70	-41.00	-21.00	-1.78	-22.39
	MaskRegulate	-12.30	-17.40	-11.50	-11.60	-18.10	-28.10	-14.70	-1.95	-14.46
	EXPlace	-8.63	-7.80	-6.65	-8.96	-9.13	-20.90	-5.29	-1.32	-8.59
	COMPAS	**-6.38**	**-2.06**	**-3.19**	**-1.93**	**-3.05**	**-12.90**	**-4.69**	**-0.64**	**-4.36**



(a) Mean paired difference in best-so-far MP-HPWL improvement.



(b) Mean best-so-far MP-HPWL improvement for COMPAS and its ASI ablations.

Figure 6: Macro-stage quality over training across eight benchmarks. (a) Mean paired difference between COMPAS and COMPAS w/o ASI; shading denotes the 95% design-level bootstrap confidence interval. (b) Mean best-so-far MP-HPWL improvement for COMPAS, COMPAS w/o ASI, and COMPAS with fixed λ.

Table 6: Mean best-so-far MP-HPWL improvement.

Method	Improvement (%)
COMPAS w/o ASI	65.26
COMPAS with fixed λ	64.53
COMPAS	**66.43**

D.3 Adaptive Constraint Dynamics

COMPAS attains lower final mean GP-HPWL than the fixed-λ ablation on seven of eight benchmark designs (Fig. 7). Fig. 8 further examines the controller dynamics through the multiplier λ and exponential moving average (EMA) of the signed macro-constraint residual. Despite the same initialization, the multiplier follows distinct trajectories across runs and benchmark designs because the accumulated constraint residuals differ. The residuals measure MP-HPWL degradation among GP-competitive placements and vary across designs and optimization runs. Consequently, the multiplier adapts to the placement outcomes encountered during training.

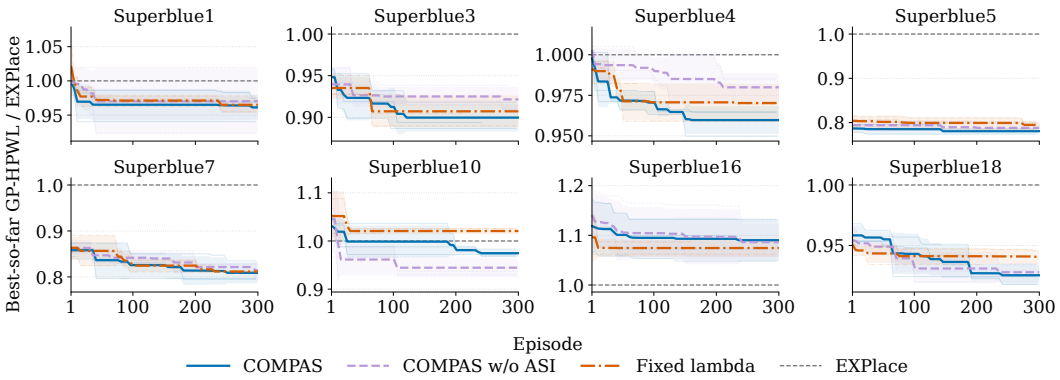


Figure 7: Convergence comparison of COMPAS, COMPAS w/o ASI, and fixed-λ. Best-so-far GP-HPWL normalized by EXPlace over 300 training episodes across eight Superblue benchmarks. Curves show the seed mean and shaded regions indicate ± 1 standard deviation. Lower is better.

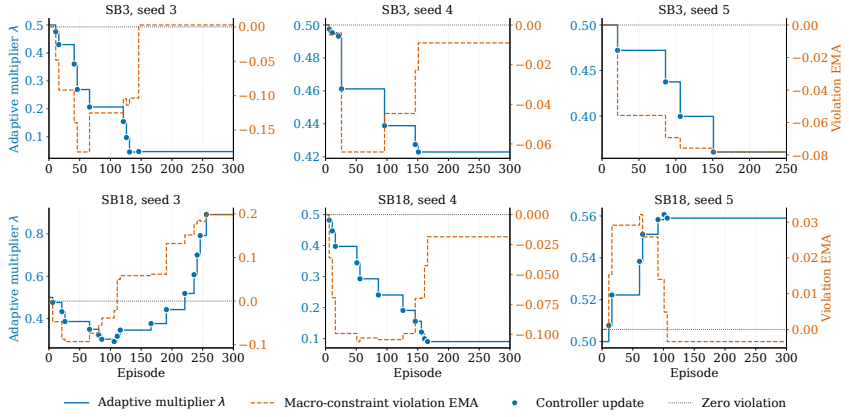


Figure 8: Adaptive macro-quality constraint dynamics across runs. Multiplier λ and EMA of the signed macro-constraint residual for three runs on Superblue3 and Superblue18. Blue markers denote controller updates. Positive residuals increase λ and negative residuals reduce it, showing adaptation across designs and over optimization.

D.4 Generalization Results

Cross-Design Generalization on ICCAD 2015 Contest Suite. Table 7 reports generalization to held-out ICCAD 2015 benchmarks. With seven held-out benchmarks, COMPAS achieves the lowest average GP-HPWL (5.75) and the best result on five of the seven benchmarks, compared with 6.05 for EXPlace. With four held-out benchmarks, COMPAS again achieves the lowest average GP-HPWL (5.87), with the best result on three of the four benchmarks. These results indicate that the improvements of COMPAS extend beyond the benchmarks used for training.

Table 7: Generalization results of GP-HPWL ($\times 10^8$) in ICCAD 2015 contest benchmarks. Dashes denote training benchmarks. † Results are reported as provided in EXPlace (Gao et al., 2026). * Results are taken from MaskRegulate (Xue et al., 2024).

Setting	Approach	SB3	SB4	SB5	SB7	SB10	SB16	SB18	Avg.
7 held-out benchmarks	MaskRegulate†	7.61	4.32	7.10	9.10	11.91	5.17	2.92	6.88
	EXPlace†	6.99	3.93	6.66	8.72	**8.63**	**4.73**	2.70	6.05
	COMPAS	**6.24**	**3.76**	**5.53**	**6.59**	10.61	4.97	**2.55**	**5.75**
4 held-out benchmarks	MaskRegulate*	–	–	–	7.99	11.55	5.16	3.04	6.94
	EXPlace	–	–	–	8.05	9.47	**4.69**	2.77	6.25
	COMPAS	–	–	–	**6.93**	**9.17**	4.82	**2.57**	**5.87**

Cross-Benchmark Generalization to ISPD 2005 benchmark suite. Table 8 compares COMPAS with the published MaskRegulate results on the corresponding ISPD 2005 evaluation protocol. Across the eight ISPD 2005 benchmarks, COMPAS reduces average GP-HPWL by 20.9% relative to the reported MaskRegulate results. We omit EXPlace from the ISPD 2005 comparison because its released pipeline is designed for ICCAD 2015 and OpenROAD benchmarks and depends on design-specific preprocessing. Extending it to ISPD 2005 would require additional implementation choices beyond the published configuration, potentially confounding a direct comparison.

Table 8: Cross-benchmark generalization to the ISPD 2005 benchmarks. Models are trained on Superblue 1, 3, 4, and 5 and evaluated on the eight ISPD 2005 benchmarks. Lower GP-HPWL is better. * Results are reported as provided in (Xue et al., 2024).

Approach	AD1	AD2	AD3	AD4	BB1	BB2	BB3	BB4	Avg.
MaskRegulate*	7.75	9.53	20.40	24.45	9.34	24.32	36.97	93.96	28.34
COMPAS	**7.28**	**8.19**	**19.25**	**17.34**	**8.93**	**13.67**	**30.36**	**74.28**	**22.41**

D.5 Ablation Study Details for Priors

As shown in Fig. 9, a conventional WireMask in place of GeoWireMask degrades placement quality across the evaluated designs, which indicates the value of informative spatial guidance. However, spatial guidance alone does not ensure continued policy improvement. COMPAS augments this guidance with downstream feedback through GP-selected reference memory and adaptive self-imitation.

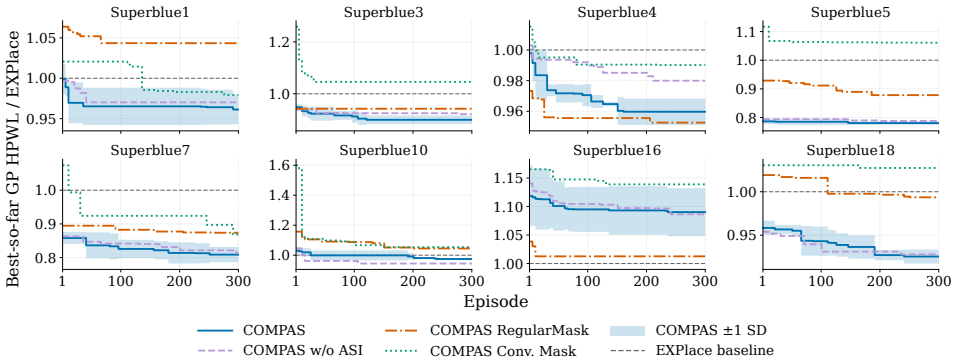


Figure 9: Best-so-far GP-HPWL normalized to the EXPlace baseline over 300 episodes on eight Superblue benchmarks. Lines show the mean over three runs per method; shading denotes ± 1 standard deviation for COMPAS. Lower is better.

D.6 Computational Cost

Table 9 shows the peak memory and runtime results of this analysis on eight designs in the ICCAD 2015 benchmark suite. Experiments were run on an AMD Ryzen Threadripper PRO 7965WX with 24 cores and 503 GiB RAM. All methods were executed sequentially under CPU-only computation. Table 10 reports the observed end-to-end elapsed time on CPU through 300 episodes on the ICCAD 2015 contest benchmarks.

Table 9: Runtime and peak memory for two-episode workflows on the ICCAD 2015 contest benchmarks. The final row reports the mean across benchmarks; lower is better.

	Runtime (s)			Peak memory (GiB)		
Design	COMPAS	MaskRegulate	EXPlace	COMPAS	MaskRegulate	EXPlace
superblue1	406.38	418.94	732.96	10.22	11.60	37.65
superblue3	484.27	534.92	644.91	11.56	12.16	33.21
superblue4	235.51	271.67	401.79	8.95	10.29	23.98
superblue5	325.76	411.51	495.04	11.27	12.61	30.59
superblue7	669.44	629.86	1351.40	18.29	18.32	55.62
superblue10	588.69	645.28	953.96	17.76	18.88	48.39
superblue16	284.31	341.16	475.74	13.35	14.75	28.82
superblue18	201.93	259.86	293.32	12.98	13.83	27.14
Mean	**399.54**	439.15	668.64	**13.05**	14.06	35.68

Table 10: Elapsed time through 300 training episodes on the ICCAD 2015 contest benchmarks on CPU.

	SB1	SB3	SB4	SB5	SB7	SB10	SB16	SB18	Avg.
Time (h)	23.29 ± 0.27	30.74 ± 0.63	11.34 ± 0.02	21.61 ± 0.16	28.05 ± 0.33	36.08 ± 0.86	26.56 ± 0.20	20.13 ± 0.30	24.73

AI USE STATEMENT

Generative AI tools were used to assist in writing polish. All methodology, experiments, results, and conclusions were developed and verified by the authors. The authors take responsibility for the final manuscript and artifacts.