

Supplementary material

A Neural Hierarchical-Matrix Preconditioner for Real-Time GPU Solves

Carl Osborne Minghao Guo Crystal Owens Wojciech Matusik
MIT CSAIL

SIGGRAPH Asia 2026 Posters (SA Posters '26). doi:10.1145/3829333.3847989

This supplement holds the training run behind the abstract’s objective comparison (§3), the transfer probe behind its generalization statement, and a picture of how local and hierarchical preconditioners move error.

S1 The probe objectives along a truncated-Kaporin run

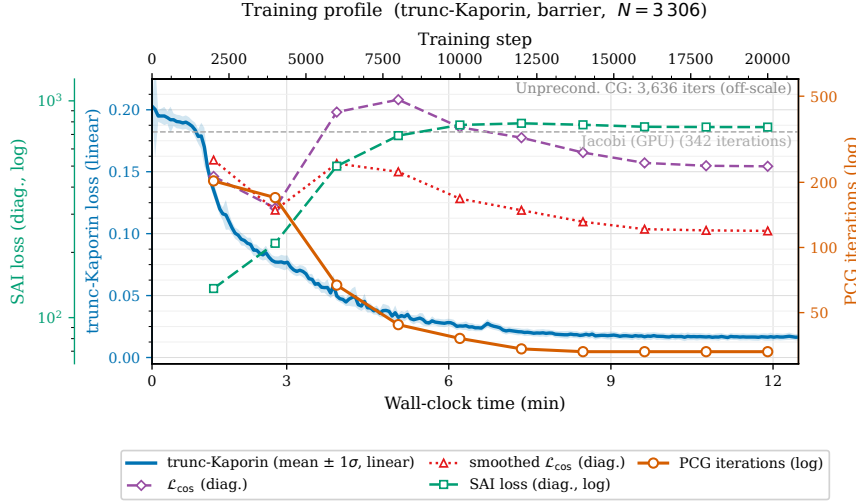


Figure 1: One truncated-Kaporin training run on the barrier-heat problem at $N = 3,306$, on one GPU. Only the truncated-Kaporin loss is trained; the probe objectives of the abstract’s §3 (“diag.”) are evaluated at each checkpoint. The two cosine losses share the truncated-Kaporin axis, and the SAI loss is the abstract’s distance loss.

Trained on its own, each probe objective does produce a real preconditioner: on this frame the distance, cosine and smoothed-cosine objectives reach 116, 91 and 68 iterations (abstract §3), against 3,636 for unpreconditioned CG and 342 for Jacobi. If they were weaker versions of the truncated-Kaporin objective, following the Kaporin run would drive them down too. It does not. As the iteration count falls to 33, the SAI loss climbs roughly elevenfold, $\mathcal{L}_{\cos}$ ends above where it started, and only the smoothed cosine falls, by about a third. The objectives are on different paths. Each rewards its own kind of preconditioner, and the path the probe losses reward is the one that cannot see the near-null modes, the λ -suppression of the abstract’s §3.

S2 What does not transfer, and why

Table 1: Transfer probe: trained at one size or family, evaluated at another. Median PCG iterations (total ms) to a true relative residual of 10^{-8} on one L40S; 5000 is the cap. “bar”: barrier-heat; “cham”: multi-chamber walls, same PDE, both at $N = 3,306$.

Regime	Train → eval	it (ms)
Same size	1,029 → 1,029	34 (4.2)
	2,072 → 2,072	24 (3.6)
	3,647 → 3,647	39 (7.4)
Larger	572 → 1,029	154 (12.1)
	1,029 → 2,072	208 (16.0)
	2,072 → 3,647	314 (24.7)
Smaller	1,029 → 572	5000 (506)
	2,072 → 1,029	5000 (512)
	3,647 → 2,072	5000 (555)
Other family	bar → cham	81 (8.5)
	cham → bar	98 (9.8)

Moving up in size costs a factor of five to nine in iterations against a network trained at the target size, moving down fails outright, and moving across problem families at $N = 3,306$ costs a factor of about two against the 44 iterations in distribution (abstract, Table 1): the preconditioner we trained does not generalize well.

That is a limit of what our network learned, not something the methods of the abstract should impose. The truncated-Kaporin objective is computed on each matrix alone, with no reference to a training set, and the network’s weights do not depend on the number of leaves, so neither is tied to a size or a geometry. What fails for us is the learned map from matrix to factor, which saw one mesh, 60–100 frames and about ten minutes of training per size. Training a network that produces these preconditioners reliably across sizes and geometries is the main open problem.

S3 How error moves across a frame



Figure 2: Residual magnitude $|r_k|$ during PCG on a 2D multiphase Poisson frame at $N = 16,384$. Top strip: density, right-hand side, pressure, velocity and the learned M . Each row is one method, labelled with its kind, shown at five iterations k on its own schedule.

Figure 2 shows, in the residual itself, how two kinds of preconditioner move error across a spatially ordered frame. A local preconditioner (Jacobi, incomplete Cholesky, a learned sparse approximate inverse) acts on each node’s neighbourhood, so an iteration carries a correction only a few cells. In those rows the residual starts as thin bands along the density interface and spreads outward from it a little at a time. A hierarchical preconditioner acts on the whole graph at once: AMG through its coarse levels, ours through tiles that couple every leaf to every other at a rank set by their separation. From the first iteration our residual is reshaped over the whole domain in blocks the size of the leaves and tiles, rather than growing out from the interface. High-contrast interfaces that error must cross are some of the problems local methods serve worst, and algebraic multigrid, which handles them, needs a per-matrix setup that puts it outside the interactive regime.