Small-o tricks

For PEPS, the parameter index α factorizes as

$$\alpha \equiv (\mathtt{x}, p, lrdu)$$

where $\mathtt{x}$ is the site index, p is the physical index and $lrdu$ are the virtual indices.

There is a fact:

$$O[\mathtt{x}](s)_{p,lrdu} = 0 \text{ if } p \neq s(\mathtt{x})$$

One thus only keeps non-zero elements in Jacobians $o[\mathtt{x}](s)_{lrdu} = O[\mathtt{x}](s)_{s(\mathtt{x}),lrdu}$. The point is in how to use these small-os' to reconstruct $O^\dagger O$ or $OO^\dagger$ and choose different implementations for a better tradeoff between memory and efficiency.

SR

$$\begin{aligned}\sum_s O^\dagger_{\alpha s} O_{s\alpha'} &= \sum_s O^\dagger[\mathtt{x}](s)_{p,lrdu} O[\mathtt{x}'](s)_{p',l'r'd'u'} \\ &= \sum_s O^\dagger[\mathtt{x}](s)_{p,lrdu} O[\mathtt{x}'](s)_{p',l'r'd'u'} \mathbf{1}[p = s(\mathtt{x})] \mathbf{1}[p' = s(\mathtt{x}')] \\ &= \sum_s \underbrace{o^\dagger[\mathtt{x}](s)_{lrdu} \mathbf{1}[p = s(\mathtt{x})]}_{\tilde{o}^\dagger_{\mathrm{SR}}} \; \underbrace{o[\mathtt{x}'](s)_{l'r'd'u'} \mathbf{1}[p' = s(\mathtt{x}')]}_{\tilde{o}_{\mathrm{SR}}}\end{aligned}$$

minSR

$$\begin{aligned}\sum_\alpha O_{s\alpha} O^\dagger_{\alpha s'} &= \sum_{\mathtt{x}} \sum_p \sum_{lrdu} O[\mathtt{x}](s)_{p,lrdu} O^\dagger[\mathtt{x}](s')_{p,lrdu} \\ &= \sum_{\mathtt{x}} \sum_p \sum_{lrdu} O[\mathtt{x}](s)_{p,lrdu} O^\dagger[\mathtt{x}](s')_{p,lrdu} \mathbf{1}[p = s(\mathtt{x})] \mathbf{1}[p = s'(\mathtt{x})] \\ &= \sum_p \sum_{\mathtt{x}} \sum_{lrdu} \underbrace{o[\mathtt{x}](s)_{lrdu} \mathbf{1}[p = s(\mathtt{x})]}_{\tilde{o}_{\mathrm{minSR}}} \; \underbrace{o^\dagger[\mathtt{x}](s')_{lrdu} \mathbf{1}[p = s'(\mathtt{x})]}_{\tilde{o}^\dagger_{\mathrm{minSR}}}\end{aligned}$$

which is exactly the small-o trick in minSR, with a mere difference that one has no need to store a $N_s \times N_p$ sample tensor. But maybe in practice one might prefer calculating a $N_s \times N_p$ sample tensor for a better dense GEMM.

enumeration order

For the small-o trick, one enumerates over p first, which replace . One can also consider other enumeration orders. replace $\sum_p$ first: One $(N_s, N_p) \times (N_p, N_s)$ GEMM with:

- $\sum_p$ first: → $\dim(p)$-many $\left(N_s, \frac{N_p}{\dim(p)}\right) \times \left(\frac{N_p}{\dim(p)}, N_s\right)$ ones
- $\sum_x$ first: → n_{site}-many $\left(N_s, \frac{N_p}{n_{\text{site}}}\right) \times \left(\frac{N_p}{n_{\text{site}}}, N_s\right)$ ones

I think one should follow the principle that:

1. Gradients should be recorded as block-sparse.
2. If you have enough memory, enumerate over the index with the smallest dimension (e.g. $\dim(p) = 2$ for qubits) for large dense GEMMs.

LGT

For gauge-invariant PEPS, the gradients are extremly sparse:

$$O[\mathtt{x}](s)_{p,lrdu} = 0 \text{ if } p \neq s(\mathtt{x}) \text{ and } lrdu \notin \text{virtual space specified by}$$

PEPS with nearest blockade

Require all virtual spaces to be a direct sum: $V_0 \oplus V_1$.

$$A_{p,lrdu} \neq 0 \text{ for}$$

p	u	l	d	r

The number of parameters is $\mathrm{O}()$