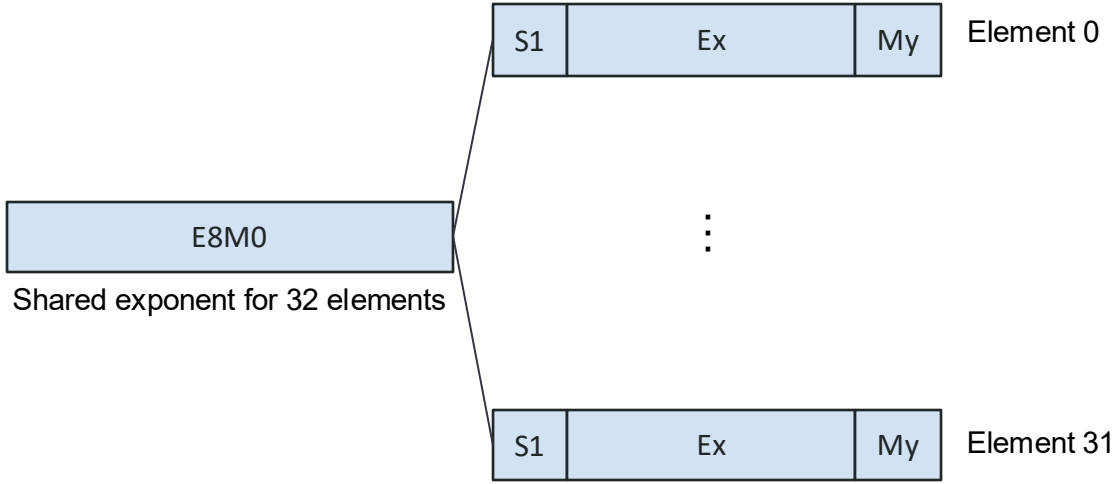
An Empirical Study of Microscaling Formats for Low-Precision LLM Training

Hanmei Yang, Summer Deng, Amit Nagpal, Maxim Naumov, Mohammad Janani, Tongping Liu, Hui Guan

Introduction to OCP Microscaling Formats (MX)

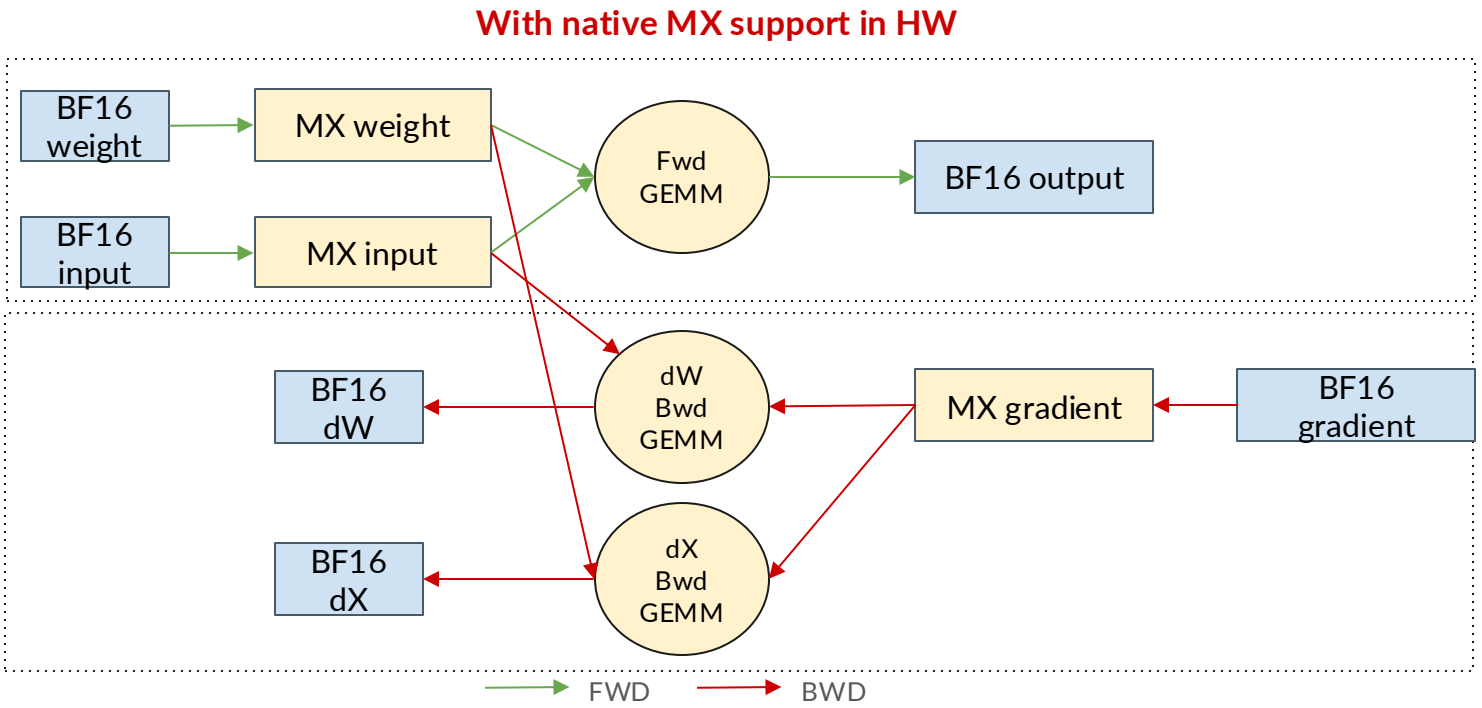


- Use block-wise microscaling
- Scale data format: E8M0
- Element data format:
  - FP8: E4M3, E5M2
  - FP6: E2M3, E3M2
  - FP4: E2M1
  - INT8

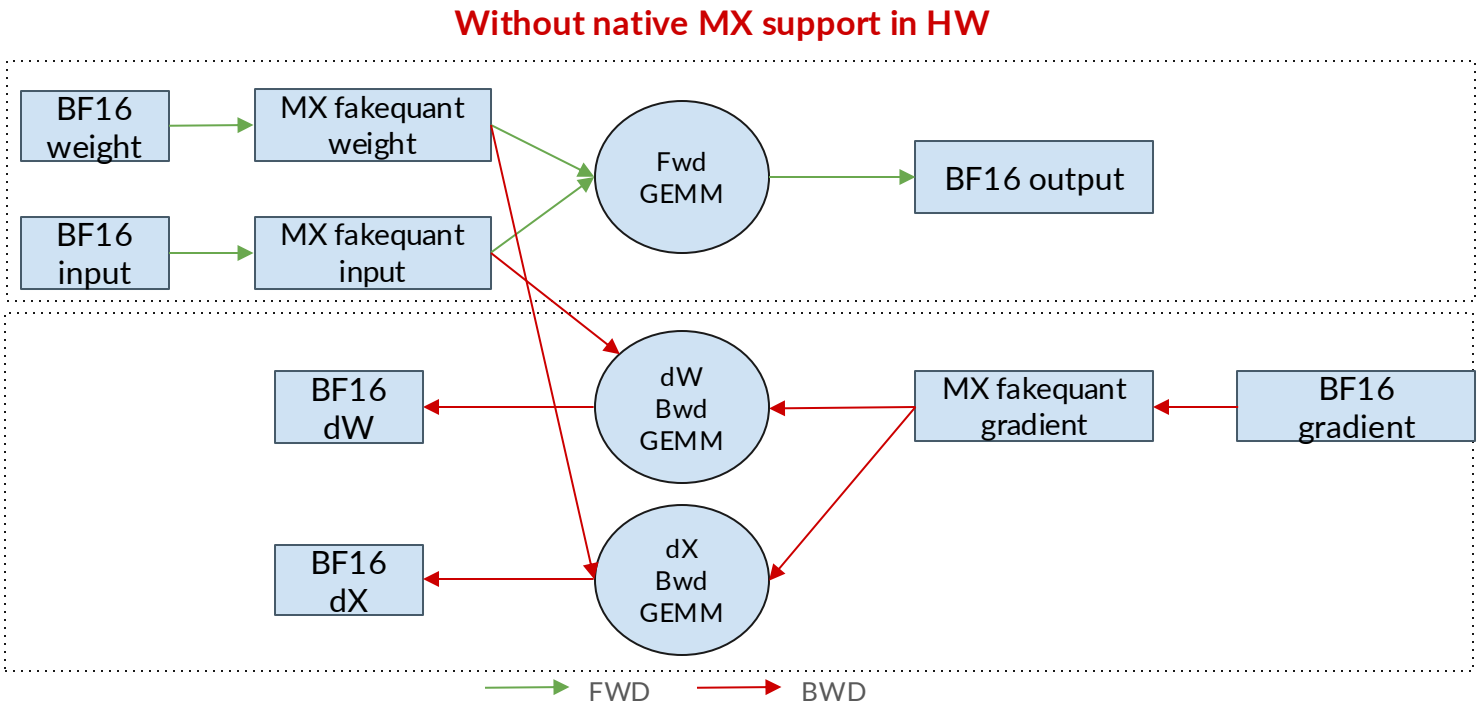
3.76x memory / transfer reduction vs. BF16

2.73x lower runtime overhead vs. BF16

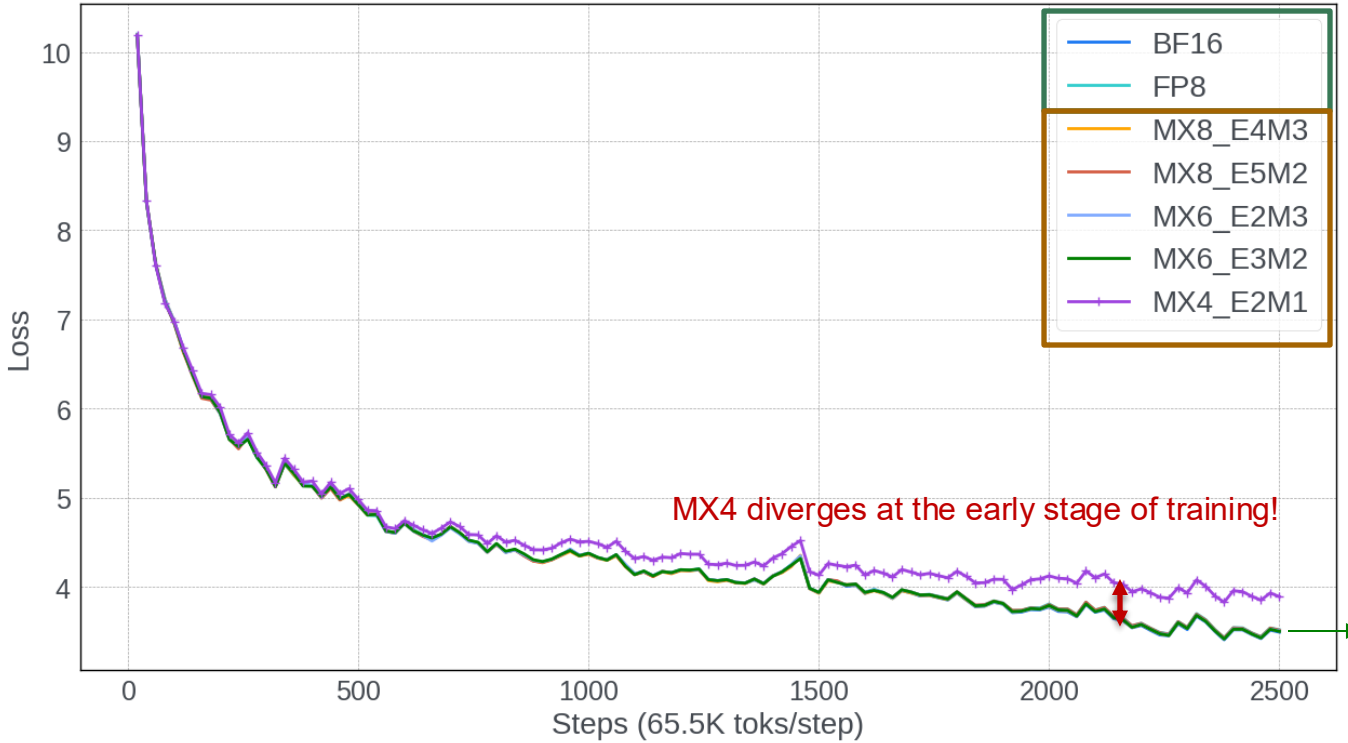
MX Quantization Workflow on Blackwell GPUs



MX Quantization Workflow on A100/H100 GPUs



Behavior of Different MX Formats During LLM Training

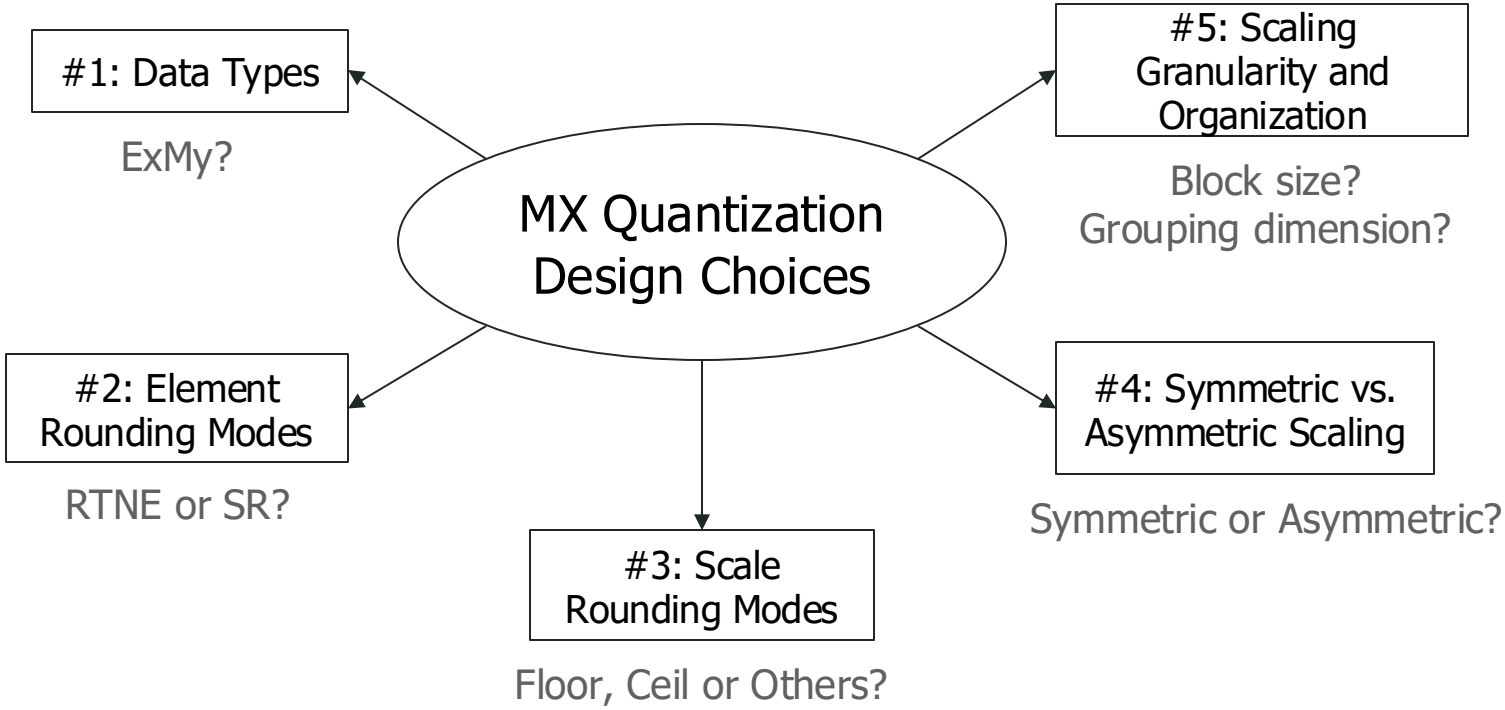


Baselines

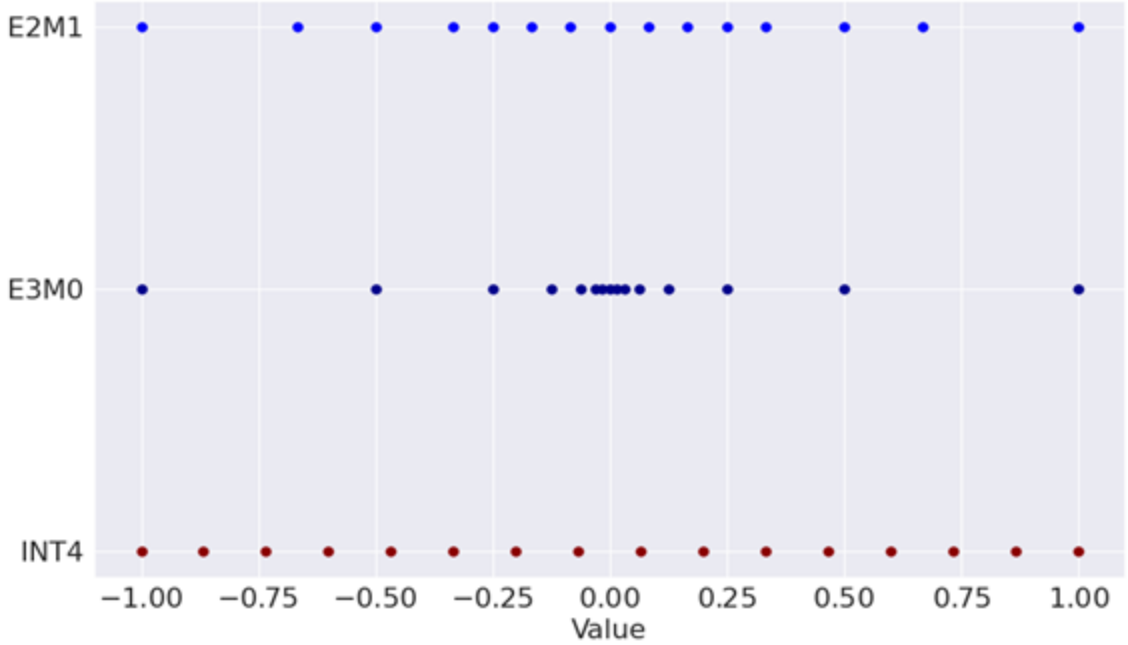
OCP MX formats

MX8 and MX6 align with baselines.

Proposed Design Space for MX Quantization



Design Choice #1: Data Types



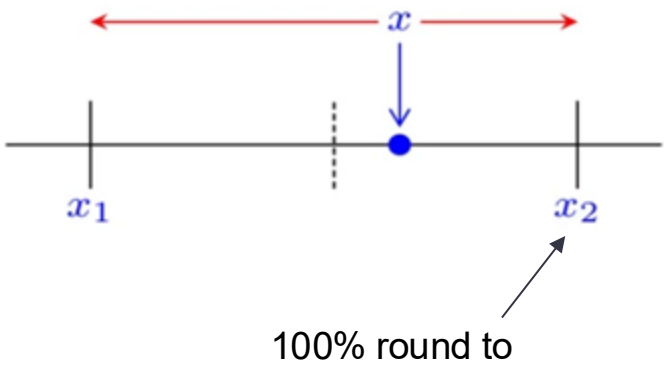
E2M1:
- OCP [1] defined MXFP4 format

E3M0:
- Larger dynamic range
- Reduced precision
- Recommended for gradients [2]

INT4:
- Better precision
- Limited dynamic range

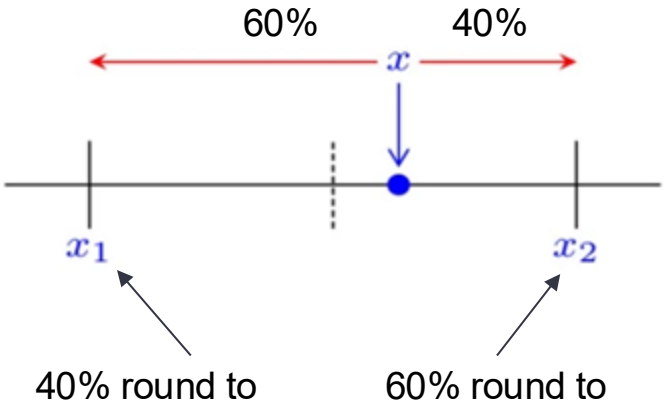
[1] OCP Microscaling Formats (MX) V1.0 Specification https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf
[2] Neural gradients are near-lognormal: improved quantized and sparse training (ICLR'21)

Design Choice #2: Element Rounding Modes



Round to Nearest Even (RTNE)

- Always rounds to the nearest deterministically.
- Default rounding mode in OCP MX specification.
- Reduces local rounding error.



Stochastic Rounding (SR)

- Rounds probabilistically based on distance.
- Introduces larger variance compared to RTNE.
- Helps preserve small-magnitude values, important for gradient quantization.

Design Choice #3: Scale Rounding Modes

Floor: scale rounding mode used in OCP Spec

$$M_{\text{hp}} = \max(|X_{\text{hp}}|)$$

$$S = 2^{\lfloor \log_2(M_{\text{hp}}) - \text{maxExp} \rfloor}$$

For E2M1, maxExp = 2, maxValue = 6

$$M_{\text{lp}} = \text{E2M1}\left(\frac{M_{\text{hp}}}{2^{\lfloor \log_2(M_{\text{hp}}) - \text{maxExp} \rfloor}}\right)$$

$$< \text{E2M1}\left(\frac{M_{\text{hp}}}{2^{\log_2(M_{\text{hp}}) - 2 - 1}}\right) = \text{E2M1}(8) = 6$$

Overflow!

Design Choice #3: Scale Rounding Modes

Ceil: avoid overflow issues

$$S = 2^{\lceil \log_2(M_{\text{hp}}) - \text{maxExp} \rceil}$$

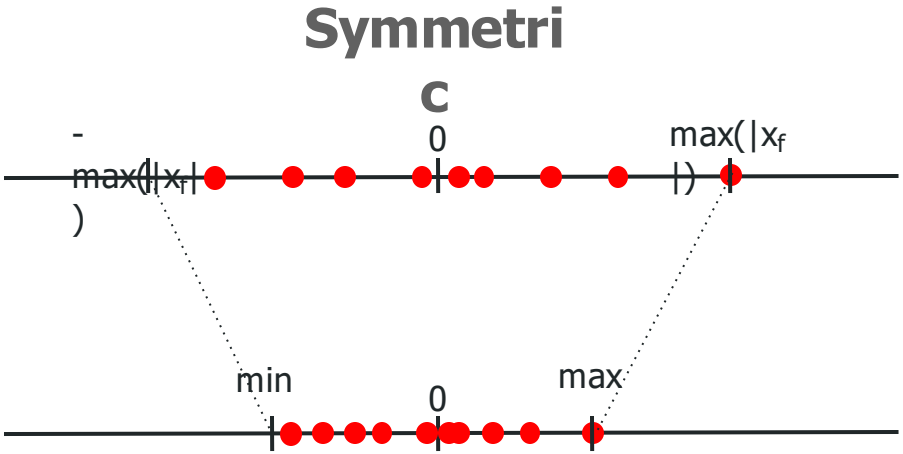
Although it prevents overflow, it may cause small values to underflow and shift to 0 due to the larger scale.

Even: round M_{hp} with RTNE before log

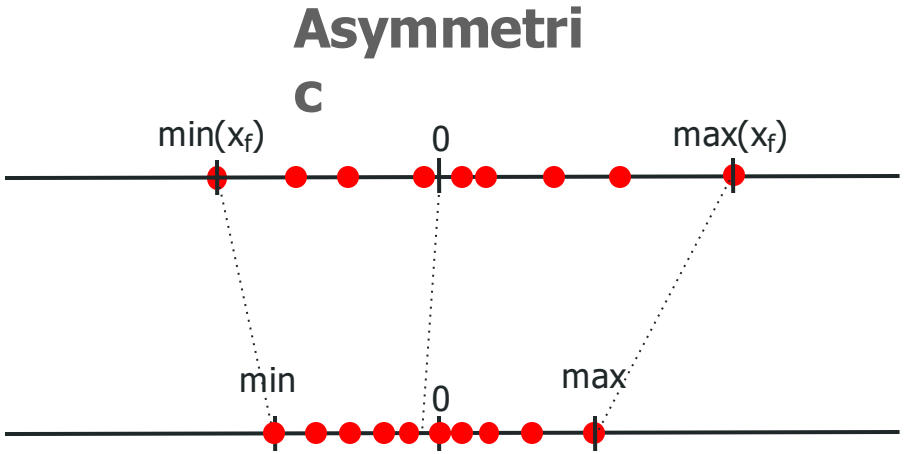
$$S = 2^{\lfloor \log_2(\text{Round}(M_{\text{hp}})) - \text{maxExp} \rfloor}$$

Trade-off between "Floor" and "Ceil"

Design Choice #4: Symmetric vs. Asymmetric Scaling



Default scaling mode in OCP Spec, simplify hardware design

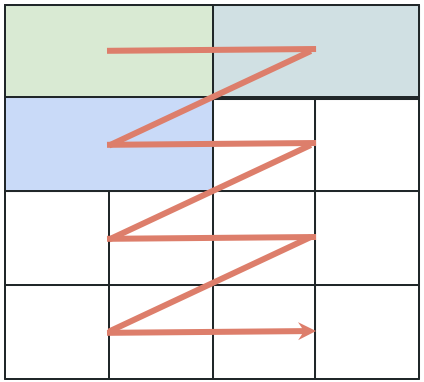


Helps when data distribution is not centered at zero

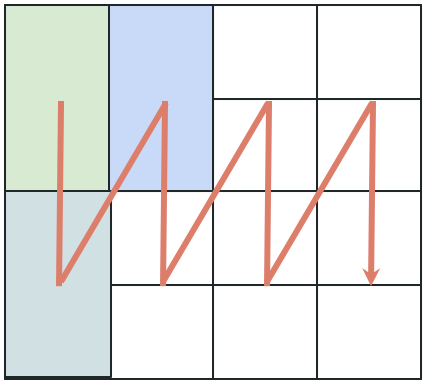
Design Choice #5: Scaling Granularity and Organization

Granularity → how many elements share one scale

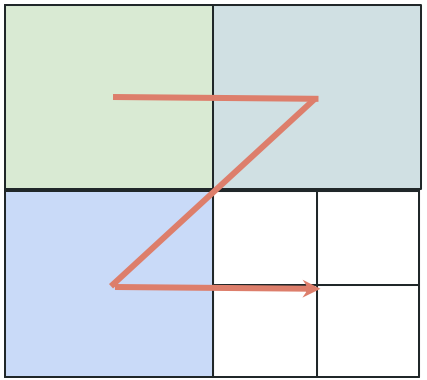
Organization → how elements are grouped across dimensions



1*2 row-wise scaling

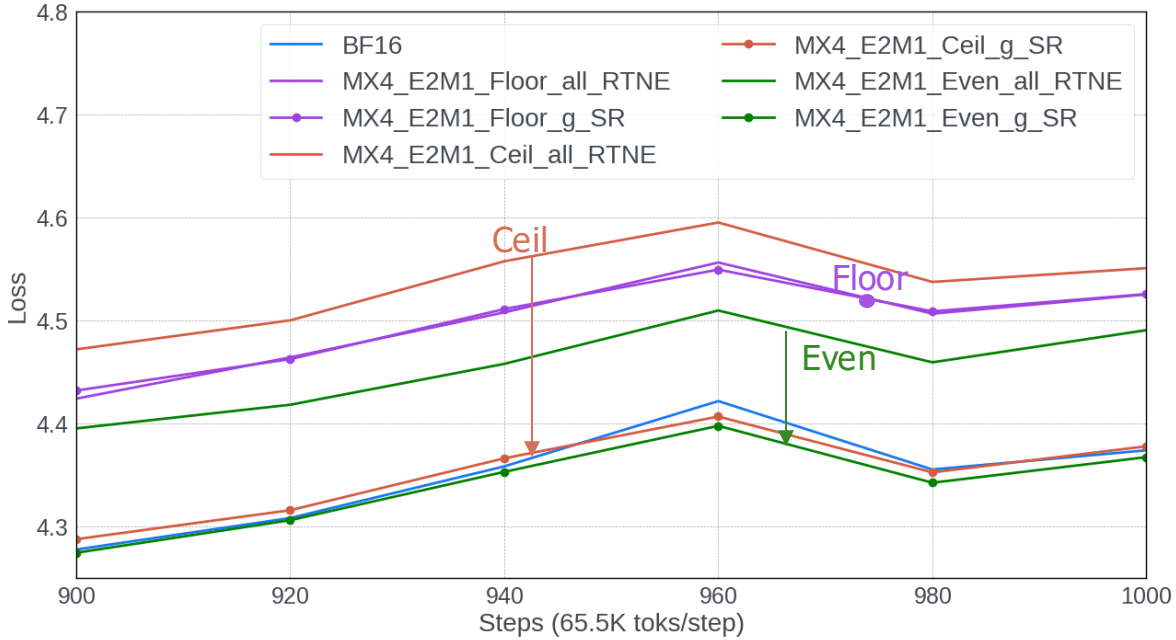


2*1 column-wise scaling



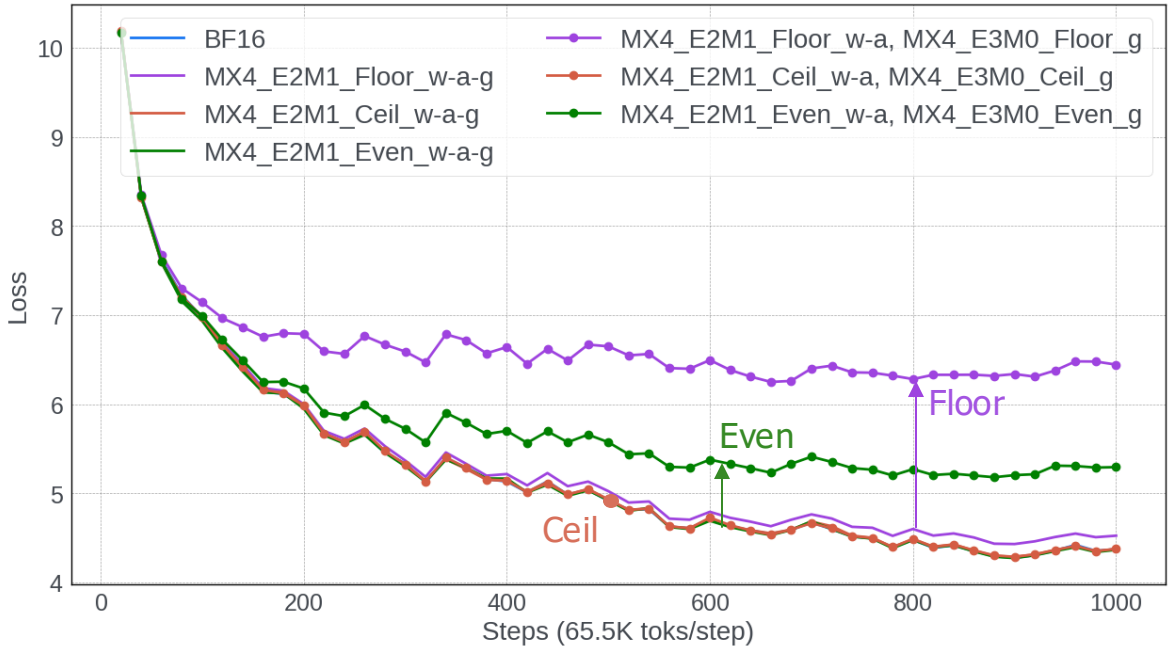
2*2 block-wise scaling

Impact of Scale and Element Rounding Modes



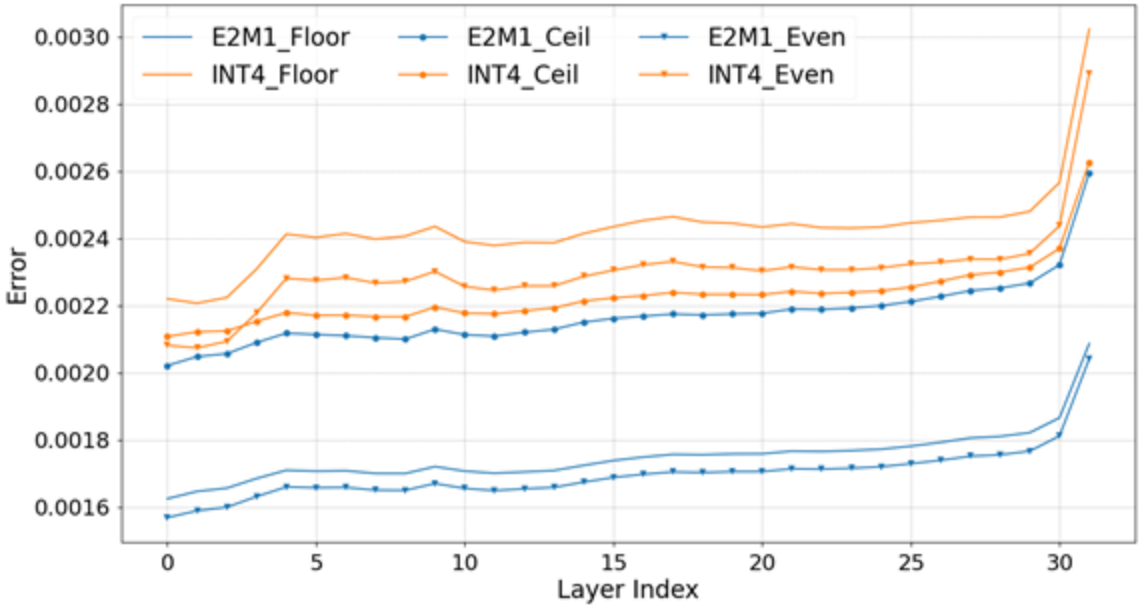
- RTNE for all tensors (weights, activations, and gradients)
  - Ceil > Floor > Even > BF16 (higher is worse)
  - Ceil suffers from underflow, leading to gradient vanishing
  - Even works the best for better balance between underflow and overflow
- When SR is applied to gradients
  - Floor > Ceil ≈ Even ≈ BF16
  - SR helps alleviates gradient vanishing for Ceil
  - Floor still faces overflow issues
- SR is less effective for weights and activations (check the paper)

Impact of E3M0



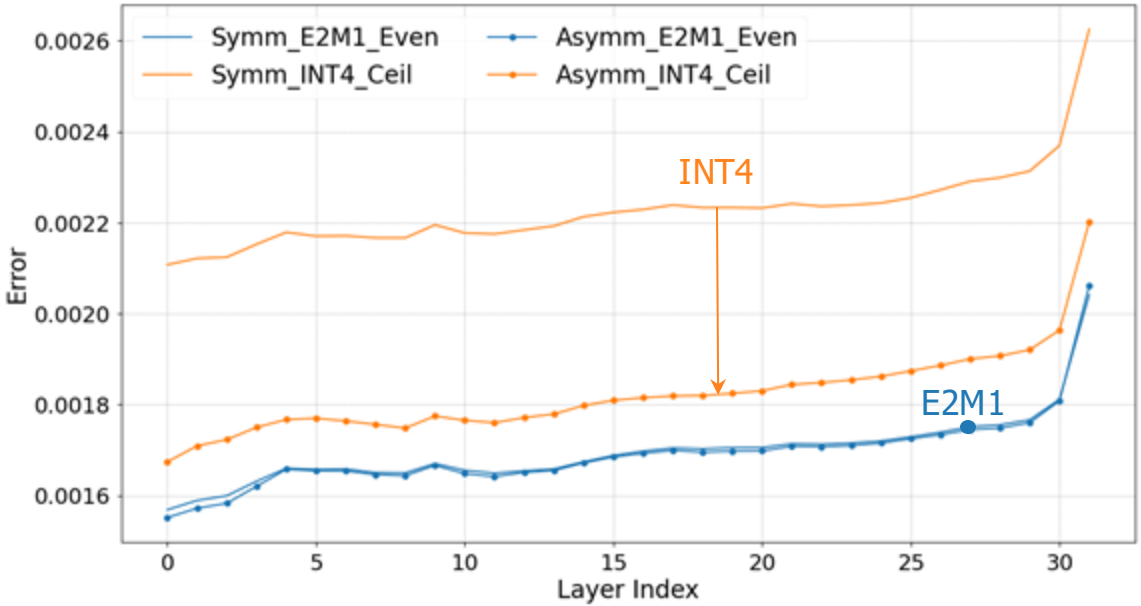
- When using E3M0 for gradients
  - Floor > Even > Ceil ≈ BF16 (higher is worse)
  - E3M0 has a wider dynamic range, causing a higher overflow penalty than E2M1.
  - That's why Floor performs poorly and Even doesn't perform as well as BF16.

Impact of INT4



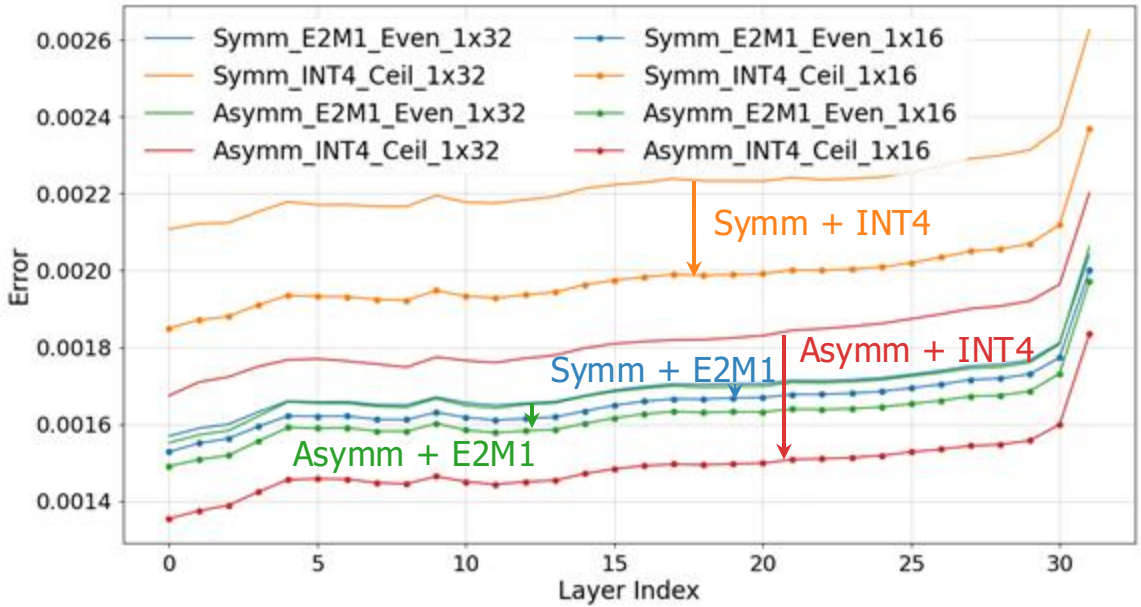
- The error is measured using Root Mean Squared Error (**RMSE**) with quantized and original tensors. We validated its correlation with training loss.
- E3M0 is excluded due to its larger error and only suits for gradients.
- Results (higher is worse)
  - E2M1: Ceil (●) > Floor (-) > Even (▼)
  - INT4: Floor (-) > Even (▼) > Ceil (●)
  - Overall: E2M1 is better than INT4

Impact of Asymmetric Scaling



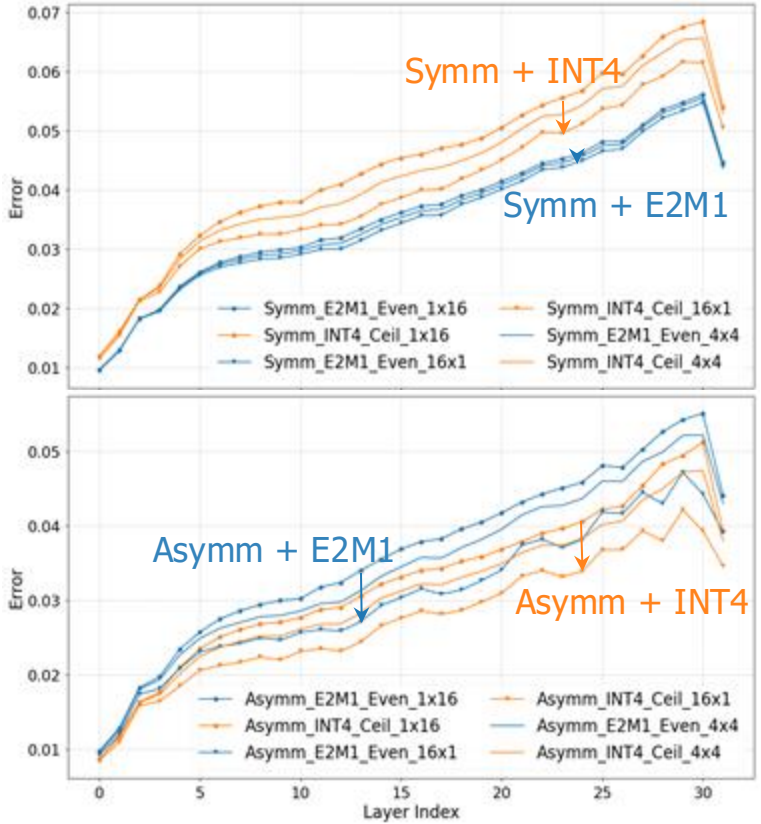
- INT4 shows significant improvement with **asymmetric scaling**, while E2M1 shows only marginal improvements.
- INT4 gains more from asymmetric scaling due to its narrower range and higher sensitivity to data skew.

Impact of Scaling Granularity



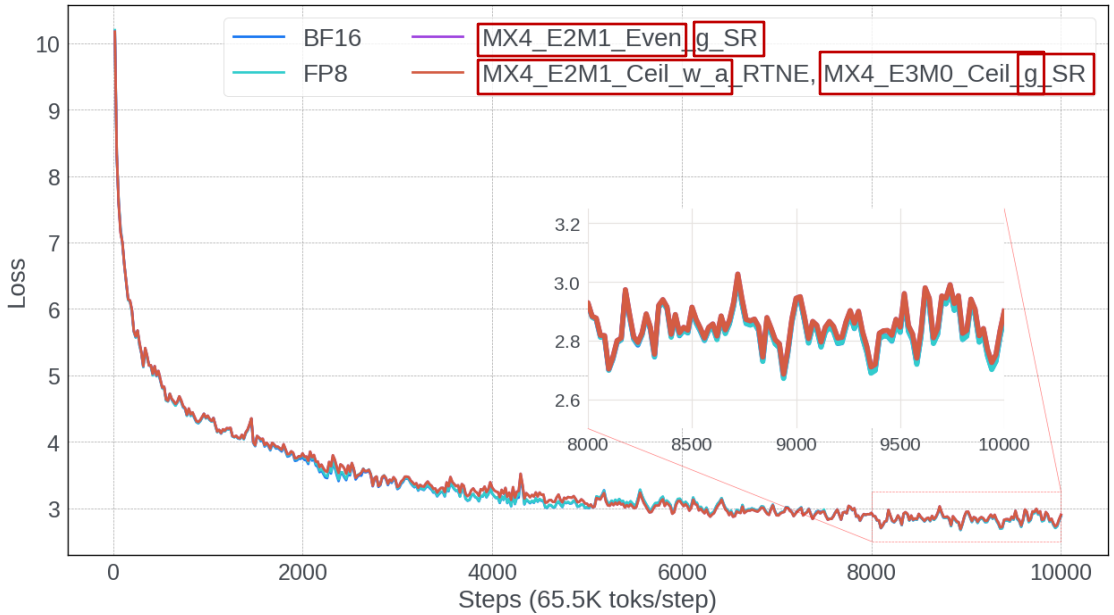
- Asymm + INT4 benefits the most from block size reduction.
- Symm + E2M1 (default MX format) benefits the least.
- **INT4** has a limited dynamic range, making **smaller block sizes** and **asymmetric scaling** more effective in capturing local variations and reducing the impact of outliers.

Impact of Scaling Organization



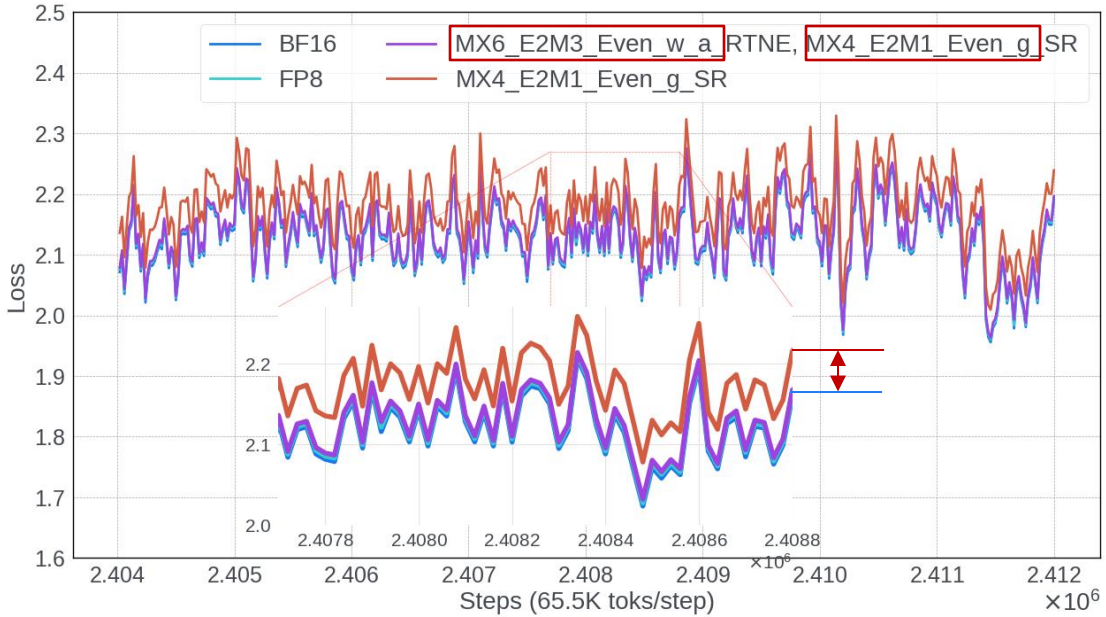
- Here, we show the results of activation quantization
  - 1x16 (●): row-wise scaling (default)
  - 16x1 (▼): column-wise scaling
  - 4x4 (-): block-wise scaling
- **Column-wise scaling** improves activation quantization.
- **Block-wise scaling** balances between precision and efficiency.
- No significant improvements of column-wise scaling in weight or gradient quantization. (check the paper)

MX Training From Scratch



- Both configurations closely track the FP8 and BF16 baselines throughout training.
- Slight divergences between 3.5K and 5K steps, but the loss curves stabilize and converge, with only a 0.02 gap by the end of training.

MX Training From a pre-trained checkpoint



- However, when resuming training from a pre-trained checkpoint, **4-bit configurations** struggle to maintain accuracy.
- Mixed-precision solution: using 4-bit for gradients and 6-bit weights and activations provides a more stable and robust solution.

Takeaways

- **Data Types and Rounding Modes**:
  - **E2M1** works well with both **Ceil** and **Even** but **Even** gives slightly better results.
  - **E3M0** is effective for *gradient*s with **Ceil**. **INT4** also requires **Ceil** to perform well.
  - **Stochastic Rounding (SR)** is essential for *gradients*, while RTNE works better for *weights* and *activations*.
- **Symmetric vs. Asymmetric Scaling**:
  - **Asymmetric scaling** benefits INT4 quantization by handling skewed data better.
  - **E2M1** exhibits minor improvements due to its wider dynamic range.
- **Scaling Granularity and Organization**:
  - Reducing block size improves quantization accuracy, with **INT4** under **asymmetric scaling** showing the largest gains.
  - **Column-wise scaling** only benefits *activation* quantization, and **block-wise scaling** provide moderate improvements and better flexibility to transpose operations during training.
- **Training Performance**:
  - **Enhanced 4-bit configurations** match FP8 and BF16 baselines when training from scratch.
  - **Mixed-precision** of **4-bit** and **6-bit** quantization maintains accuracy for resumed training.

Limitations and Future Work

- Current 4-bit quantization requires 6-bit precision for weights and activations to maintain stability in resumed training. → Improve 4-bit methods to eliminate the need for higher-precision fallback.
- Experiments are conducted on relatively short training runs. → Increase the number of tokens in the experiments to assess convergence and stability over time.
- Evaluation is limited to a single LLM architecture. → Generalize the approach by testing on more architectures.