Why does the NALU perform great in the original paper? We have contracted the authors of the original paper, unfortunately they have been unable to provide any specific details about the models or the experiments. As such, we firmly believe our implementation and experiments are identical to theirs. Our intuition about NALU is that convergence only happens for a luck initialization. In the original paper no analysis of "success rate" was done, and we thus believe their results are heavily cherry-picked. However, as we don't think such speculation is appropriate in a paper we have omitted that details. *Why is the NALU worse than the linear model?* The NALU shares weights between the NAC_+ and $\mathrm{NAC}_\bullet$ layers. Compared to a NALU with separated weights, we have observed that weight sharing makes addition harder but multiplication feasible when a gating mechanism is present. *Are the experiments designed to show a disadvantage with NALU?* No, our experiments are identical to those in the original paper. The parameters that we vary are those that are not stated in the original paper.

Is the comparison between NMU and $\mathrm{NAC}_\bullet$ unfair? Maybe, it is true that our NMU model, unlike $\mathrm{NAC}_\bullet$, does not support division. On the other hand our model can understand inputs in the negative range. To make a more fair comparison, we have added two additional models. A variant called $\mathrm{NAC}_{\bullet,\sigma}$ that only supports multiplication, by constraining the weights with $W = \sigma(\hat{W})$ in $\mathrm{NAC}_\bullet$. And a variant of $\mathrm{NAC}_{\bullet,\sigma}$ that uses linear weights and bias regularization, identically to our NMU model, this model is called $\mathrm{NAC}_{\bullet,\mathrm{NMU}}$. Figure 1 shows that both variations improve upon the original $\mathrm{NAC}_\bullet$ model, but not enough to be better than the NMU model. *The conclusion currently states the advantaged of NAU and NMU without mentioning their drawbacks.* Good point, we have fix that. Although, please note that the original paper also states that division does not extrapolate well. $\mathrm{NAC}_\bullet$ *supports* $\sqrt{x}$ *and* x^2*, does NMU support those?* Good point. NMU supports x^2, in the same way as $\mathrm{NAC}_\bullet$ does. It is just an easier version of multiplication. NMU does not support $\sqrt{x}$, we will mention that in the revision.

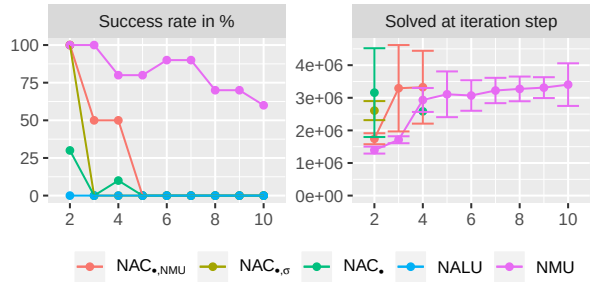

Figure 1: Shows effect of increasing hidden size, thus adding redundant hidden units. Sparsity error excluded due to space constraints.

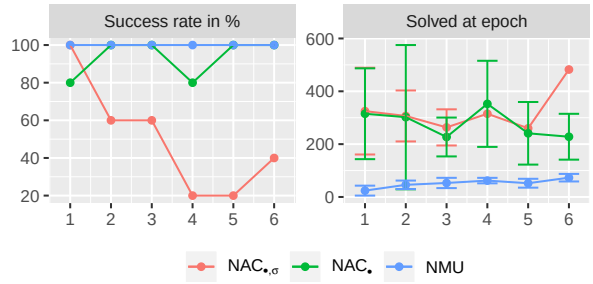

Figure 2: Shows effect of the number of mnist hidden outputs, at sequence length 9, trained on sequence length 2. Sparsity error excluded due to space constraints.

More experiments We added experiments that tests the effect having redundant hidden units, this is important in applications where the correct number of hidden units is unknown. Additionally, we added the MNIST experiment from the original paper, although we test on multiplication instead of addition. This experiment evaluates how well a larger network can back-propagate though the $\mathrm{NAC}_\bullet$ or NMU layer. This experiment shows that the NMU allows for more than 10x faster learning. *The novelty is only limited to the theoretical analyses of NALU and a new parameterization.* Our comparison should be on the baselines already established by the NALU paper, we thus don't want to invent new tasks. We hope that the additional experiments will convince you that our contributions are worth of publication.

If an initialization scheme results in $E[z_{h_l}] \neq 0$, does that mean the optimization would be necessarily difficult? Not "necessarily", $E[z_{h_l}] = 0$ is a desirable not a theoretical necessarily. We use this desirable to inspire our model choice, and validate this choice in our experiments. Hopefully, our new comparison with $\mathrm{NAC}_{\bullet,\mathrm{NMU}}$ will convince you that this is a sensible desirable. We will make it more explicit that this is only a desirable in our revision.

Other questions. *Would weight clipping have saturation issues?* No, because NMU is linear in w_i, the gradient is not affected by w_i. Gradients are derived in Appendix A. *Does regularization in the NALU help?* $\mathrm{NAC}_{\bullet,\mathrm{NMU}}$ contains the regularization from NMU, but remains worse than NMU. There is no reason to believe that the added complexity of gating in NALU would make it better. *Why did you chose* λ_{bias} *in this way?* We found that learning interpolation and then sparsify to learn extrapolation are two different optimization problems. Only the latter, requires regularization. In our revision the bias regularizer is now scaled with spline function (shape similar to ReLU6). We explain this better in our revision. *Is the parameter sparsity a necessity?* For multiplication it is a requirement. For addition and subtraction it helps with interpretability. *Are the assumptions of splitting the NALU worth the effort?* NALU has its own issues, regarding the gating mechanism. However, we believe that before the gating issue can be solved the subunits needs to converge consistently. We hope future research will provide a solution to the gating mechanism issues present in NALU. *Does the* NAC_+ *need fixing?* Our NAU converges faster, uses half the parameters, and bias towards discrete weights. We believe these contributions are noteworthy, but agree that they are minor.

Minor typos and comments Thanks for noticing, these issues have been fixed in our revision.