A Complex network

A simpler network that solves modular addition problem can be phrased using complex weights. This structure would also be more friendly to physicists. The complex solution takes form

$$W^{(1)}_{kn} = \begin{pmatrix} e^{2\pi i \frac{k}{p} n_1 + i\varphi^{(1)}_k} \\ e^{2\pi i \frac{k}{p} n_2 + i\varphi^{(2)}_k} \end{pmatrix}, \qquad n = (n_1, n_2) \tag{20}$$

$$W^{(2)}_{qk} = e^{-2\pi i \frac{k}{p} q - i\varphi^{(3)}_k}, \tag{21}$$

We can take quadratic activation function that simply squares the preactivations. The first preactivation and activation are given by

$$h^{(1)}(n, m) = e^{2\pi i \frac{k}{p} n + i\varphi^{(1)}_k} + e^{2\pi i \frac{k}{p} m + i\varphi^{(2)}_k}, \tag{22}$$

$$z^{(1)}(n, m) = e^{2\pi i \frac{k}{p} 2n + i\varphi^{(1)}_k} + e^{2\pi i \frac{k}{p} 2m + i\varphi^{(2)}_k} + 2e^{2\pi i \frac{k}{p}(n+m) + i(\varphi^{(1)}_k + \varphi^{(2)}_k)}. \tag{23}$$

The final activation is given by

$$h^{(2)}(n, m) = \sum_{k=1}^{N} \Big(e^{2\pi i \frac{k}{p}(2n-q) + i(\varphi^{(1)}_k - \varphi^{(3)}_k)} + e^{2\pi i \frac{k}{p}(2m-q) + i(\varphi^{(2)}_k - \varphi^{(3)}_k)} \tag{24}$$

$$+ \quad 2e^{2\pi i \frac{k}{p}(n+m-q) + i(\varphi^{(1)}_k + \varphi^{(2)}_k - \varphi^{(3)}_k)} \Big). \tag{25}$$

Similarly setting

$$\varphi^{(1)}_k + \varphi^{(2)}_k - \varphi^{(3)}_k = 0 \tag{26}$$

yields the constructive interference for the output supported on $(n + m - q) = 0 \bmod p$.

B Other activations

Remarkably, the weights equation 6-equation 7 also solve the modular addition problem for networks equation 1-equation 2 with other activation functions. That is, the function

$$f(x) = \frac{1}{D\sqrt{N}} W^{(2)} \phi\left(W^{(1)} x\right) \tag{27}$$

approximates the δ-function concentrated on the modular addition problem. This also holds for the generalizations discussed in the main text. We do not have an analytic proof of this fact, so we provide the evidence in Fig. 5.

C Dynamics

In this Section we introduce an empirical measure that quantifies the feature learning for the modular addition task. To define such measure we turn to the exact solution equation 6- equation 7. We will utilize the fact that periodic weights are peaked in Fourier space, while random weights are not.

To define the measure of feature learning, we first transform the weights $W^{(1)}_{nk}$ to a Fourier space with respect to index n. Denote the transformed weights $\tilde{W}^{(1)}_{\nu k}$. If the weights are periodic, then Fourier-transformed weights are *localized* in ν, *i.e.* for most values of ν we have $\tilde{W}^{(1)}_{\nu k} \approx 0$ except for a few values determined by the frequency $\frac{2\pi}{p} k$. At initialization, when the weights are random the Fourier-transformed weights are *delocalized*, *i.e.* will take roughly equal values for any ν.

We introduce a measure of localization known as the inverse participation ratio (IPR). It is routinely used in localization physics Girvin and Yang (2019) as well as network theory Pastor-Satorras and Castellano (2016). We define IPR in terms of the normalized Fourier-transformed weights

$$\text{IPR}_r(k) = \sum_{\nu=1}^{D} |\tilde{w}^{(1)}_{\nu k}|^{2r}, \qquad \text{where} \qquad \tilde{w}^{(1)}_{\nu k} = \frac{\tilde{W}^{(1)}_{\nu k}}{\sqrt{\sum_{\nu=1}^{D} (\tilde{W}^{(1)}_{\nu k})^2}}, \tag{28}$$

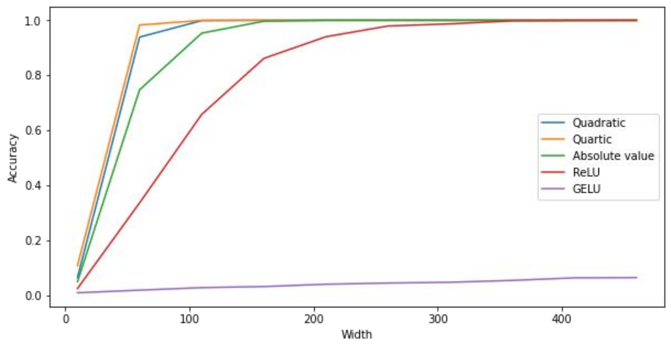


Figure 5: Accuracy for various activation functions. Test accuracy vs. width for different activation functions for $f(n, m) = n + m \bmod p$. The weights are given by equation 6-equation 7. GELU activation eventually reaches 100% accuracy, but at very large width.

and r is a parameter traditionally taken to be 2. It follows from the definition that $\mathrm{IPR}_1(k) = 1$ for any k. Unfortunately, $\mathrm{IPR}_r(k)$ is defined per neuron. We would like a single measure for all of the weights in a given layer. Thus, we introduce the average IPR

$$\overline{\mathrm{IPR}}_r = \frac{1}{N}\sum_{k=1}^{N} \mathrm{IPR}_r(k)\,. \tag{29}$$

Larger values of $\overline{\mathrm{IPR}}_r$ indicate that the weights are more periodic, while the smaller values indicate that the weights are more random.

We plot $\overline{\mathrm{IPR}}_2$ as a function of time in Fig. 6. It is clear that there is an upward trend from the very beginning of training. Onset of grokking is correlated with the sharp increase of rate of IPR growth.

D Some other modular functions

The architecture equation 4 can also learn modular multiplication, however in that case we have to convert the products into sums using logarithms over $\mathbb{Z}_p$. We will discuss the modular multiplication elsewhere.

Broadly speaking, a bivariate modular function is a $p \times p$ table where each entry can take values between 0 and $p-1$. There are p^{p^2} such tables. Grokking is not possible on the overwhelming majority of such functions, because this set includes placing random integers in each entry of the table. Such functions can be represented as modular polynomials of a sufficiently large degree. Some modular functions, namely the ones that involve addition *and* multiplication, *and* are not of the form $\tilde{f}$ are substantially harder to learn. They require more data, more time and do not always yield 100% test accuracy after grokking. One particularly interesting example was found by Power et al. (2022), $f(n, m) = n^3 + nm^2 + m$, which does not generalize even for $\alpha > 0.9$, both for transformer and MLP architectures. Some examples are discussed below. It is not clear how to predict which functions will generalize and which will not given an architecture.

To summarize, a general polynomial of degree 1 is easy to learn and the corresponding MLP is analytically solvable and interpretable, a general polynomial of degree 2 is difficult to learn and is not solvable, while a general polynomial of degree 3 cannot be learnt within the scope of methods we have tried.

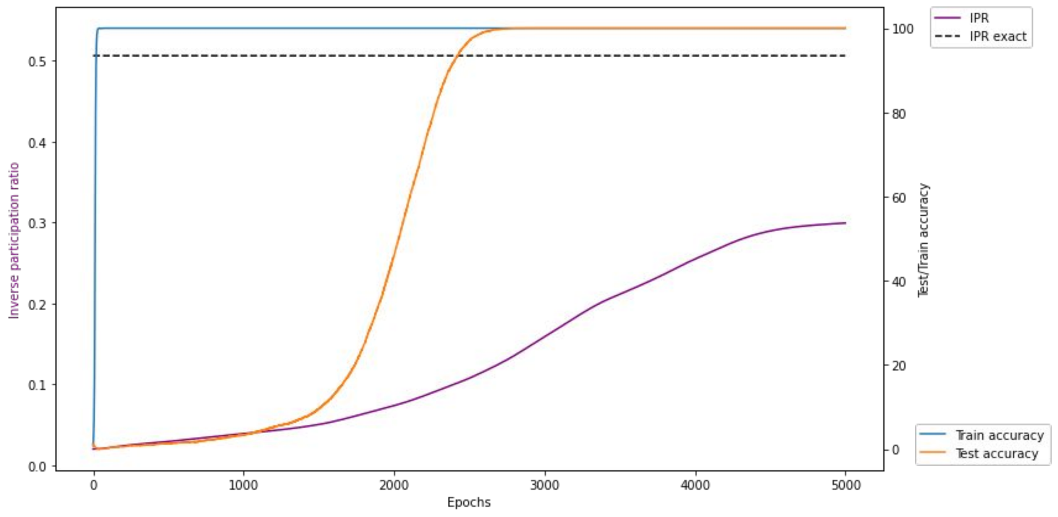


Figure 6: Inverse participation ratio. IPR plotted against the dynamics (under AdamW) of train and test accuracy. Empirically, we see 4 regimes: (i) early training when IPR grows linearly and slowly; (ii) transition from slow liner growth to fast linear growth. This transition period coincides with grokking; (iii) fast linear growth, that starts after 100% test accuracy was reached; (iv) saturation, once weights became periodic. The dashed line indicates $\overline{\mathrm{IPR}}_2$ for the exact solution equation 6-equation 7. The gap between the two indicates that even in the final solution there is quite a bit of noise leading do some mild delocalization in Fourier space. More training and more data helps to reduce the gap.

Next we show a few examples of the modular functions for which the exact solutions discussed in the main text apply.

- $f(n,m) = n^2 + m^2 \mod p$. Full solution is available and gives 100% accuracy. The first layer weights are given by

$$W^{(1)}_{kn} = \begin{pmatrix} \cos\left(2\pi\frac{k}{p}n_1^2 + \varphi^{(1)}_k\right) \\ \cos\left(2\pi\frac{k}{p}n_2^2 + \varphi^{(2)}_k\right) \end{pmatrix}, \qquad n = (n_1, n_2), \tag{30}$$

  while the second layer weights remain unmodified.

- $f(n,m) = (n+m)^2 \mod p$. The weights in the first layer are unmodified, while the weights in the second layer are given by

$$W^{(2)}_{qk} = \cos\left(-2\pi\frac{k}{p}q^{\frac{1}{2}} - \varphi^{(3)}_k\right). \tag{31}$$

  Note that $q^{\frac{1}{2}}$ must be understood in the modular sense, that is $r = q^{\frac{1}{2}}$ is a solution to $r^2 = q \mod p$.

- $f(n,m) = nm$. We do not have an analytic solution. The activations are presented in Fig. 9
- $f(n,m) = n^2 + m^2 + nm \mod p$. We do not have an analytic solution. This generalization on this function never reaches 100% unless most of the data is utilized, $\alpha > 0.95$. See the learning curve in Fig. 10. Note that although generalization accuracy is very high: $\approx 97\%$, there is a large gap between train and test loss. This is to be contrasted with Fig. 2, where the gap disappears over time.
- $f(n,m) = n^3 + nm^2 + m$. We do not have an analytic solution. The generalization never rises above 1%. See the learning curve in Fig. 10.

We show the corresponding activations on Fig. 7 - Fig. 9

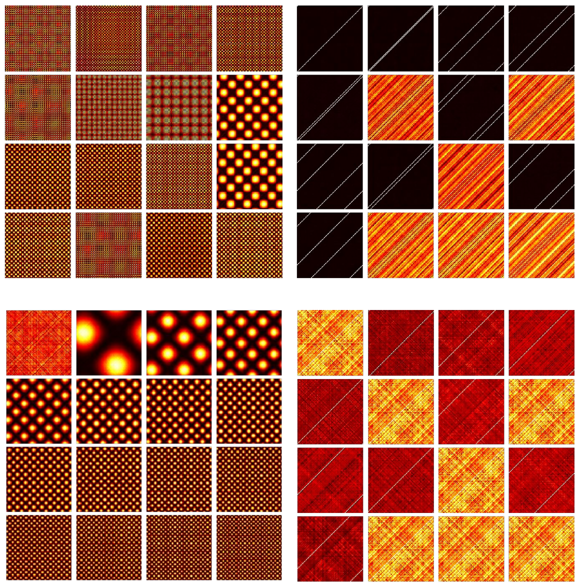


Figure 7: **Top**: Preactivations $h_k^{(1)}$ and $h_q^{(2)}$ found by the AdamW for $f(n,m) = (n+m)^2 \bmod p$. Note that $h_k^{(1)}$ is the same as for $f(n,m) = (n+m) \bmod p$ as expected. **Bottom**: Analytic solution for the same function. Note that since square root is *not* invertible – because it has two branches – the accuracy of analytic solution is $\approx 51\%$. It can be clearly seen in the form of $h_q^{(2)}$: there are 4 activation lines in the top plots and only 2 in the bottom. Each pair corresponds to a branch of square root. The noisy preactivations $h_q^{(2)}$ correspond to the values of q that cannot be represented as $(n+m)^2 \bmod p$.

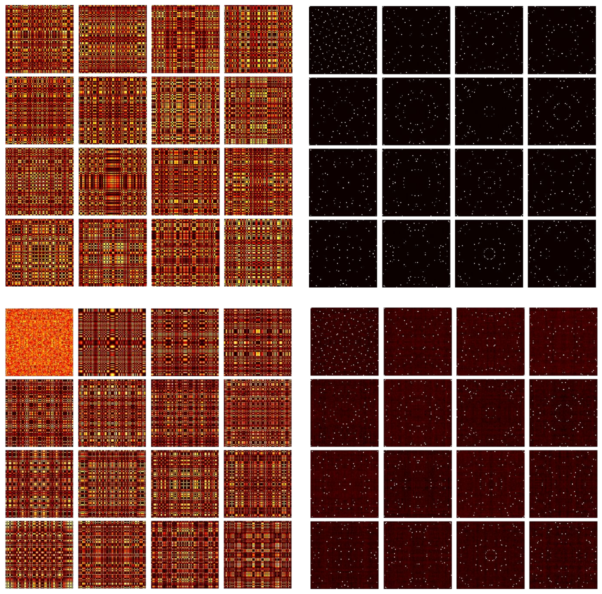


Figure 8: **Top**: Preactivations $h_k^{(1)}$ and $h_q^{(2)}$ found by the AdamW for $f(n,m) = n^2 + m^2 \bmod p$. **Bottom**: Analytic solution for the same function. Both solutions have 100% accuracy.

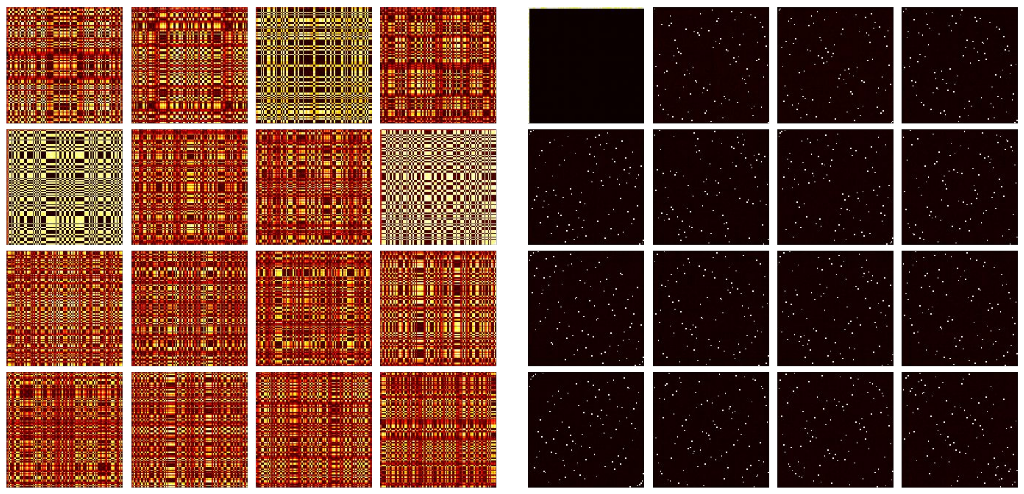


Figure 9: Preactivations $h_k^{(1)}$ and $h_q^{(2)}$ found by the AdamW for $f(n,m) = nm \bmod p$.

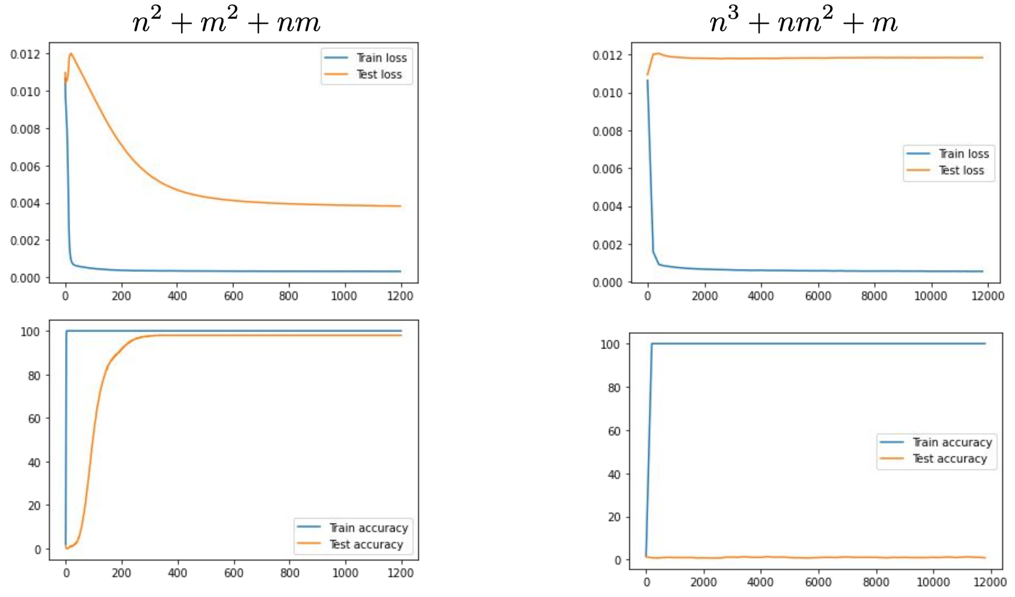


Figure 10: The learning curves for $f(n,m) = n^2 + m^2 + nm \bmod p$ and $f(n,m) = n^3 + nm^2 + m \bmod p$ at $\alpha = 0.73$ and $\alpha = 0.9$ correspondingly. Note the gap between train and test loss in the former case. Although test accuracy is almost 100%, it is clear that the network did not grok all the right features.

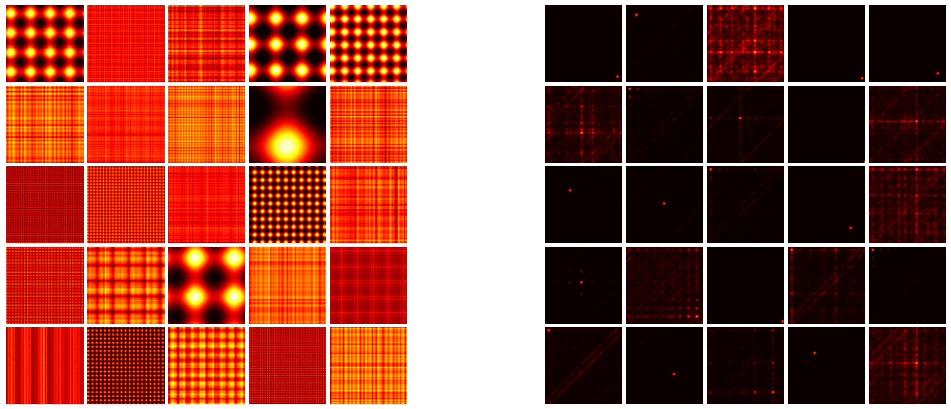
E Attention and Transformers

In this Section we empirically investigate grokking in a single-layer Transformer as well as the pure attention layers.

E.1 TRANSFORMER

Grokking was initially discovered in two-layer transformerPower et al. (2022), while the periodic structure in MLP activations was found in the case of a single-layer transformer Nanda and Lieberum (2022). Here we slightly simplify the setup by using the MSE loss (showing in passing that slingshots Thilak et al. (2022) are not necessary for the generalization to occur).

We reproduce the periodicity of the MLP activations, see Fig. 11 as expected. We also note that embedding and un-embedding matrices generically do not develop structure in Fourier space. This indicated that periodic structure is universal, while it's realization in terms of weights is not. This further underlines the utility of the analytic solution described in the present work.



(a) Activations in the MLP layer (b) Activations in the MLP layer in Fourier space

Figure 11: Activations (and their Fourier transforms) in the MLP layer of a single-layer transformer architecture. Note that the activations are periodic exactly the same way as in the MLP case, while their realization in terms of weights is different.

E.2 ATTENTION-ONLY "TRANSFORMER"

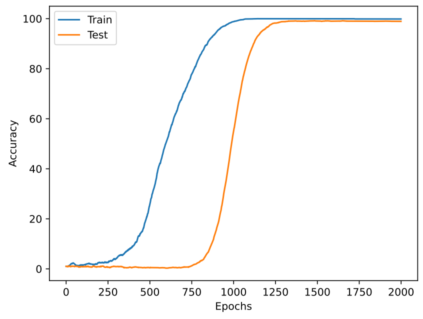
Based on the main body of the paper, it is tempting to conclude that the computations relevant for modular addition are always done in the MLP layers. We are going to challenge this conclusion in the present Section by removing the MLP layers from the transformer architecture. Indeed, in the main text we showed that MLP alone can solve the modular arithmetic problems. Consequently, adding more non-linearity via attention is not necessary: if anything, attention only gets on the way making the optimization more difficult. In all cases when MLP is present periodic activations appear there. Now we will ask the opposite question: can attention alone, being a non-linear operation, ensure generalization and grokking on arithmetic tasks?

Experiments reveal that when generalisation occurs, the dynamics shows the same characteristic features as MLP discussed in the main text and Transformers discussed in the literature as well as the present section. Fig.12.

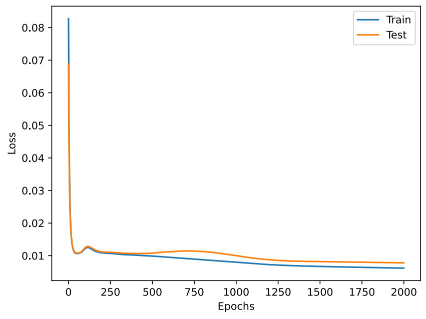
We make, however, a surprising observation: generalization on modular addition is only possible if enough attention heads are present. There is a headcount-driven grokking transition at *fixed* number of parameters. This transition is present in a single-layer linear and softmax Attention, as well as in two-layer softmax Attention.

We present the results in Fig. 13, where the final train/test accuracy[4] are plotted as functions of number of heads. The experiments are done for $p = 97$ and $n_{embd} = 128$. It is clear from Fig. 13

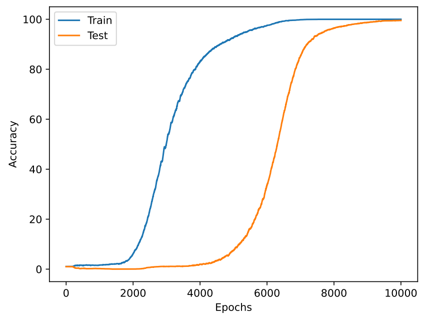
[4]With early stopping used if needed



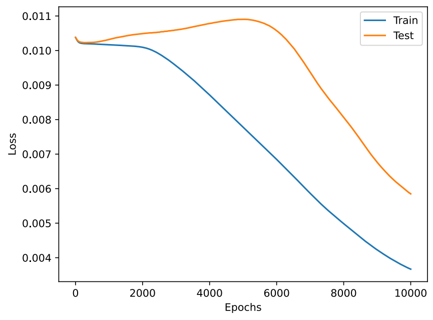
(a)



(b)



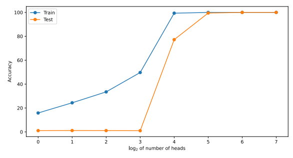
(c)



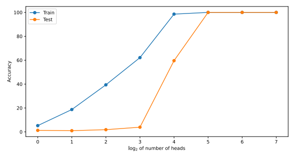
(d)

Figure 12: (a),(b): Accuracy and Loss for linear attention layers on modular addition. (c),(d): Accuracy and Loss for Softmax attention. In both cases dynamics shows the same signatures as in the MLP case.

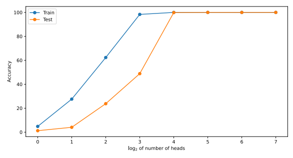
that when the number of heads is small, Attention-only network fails to approximate the training set well.



(a) Linear single-layer attention



(b) Softmax single-layer attention



(c) Softmax two-layer attention

Figure 13: In all cases we observe a generalization jump driven by the number of heads. In the experiments the number of parameters is held fixed and only the number of heads (and, correspondingly, head size) are changed.