the computation process during a forward pass: one embedding vector, a so-called *latent,* per input token $x_1, \ldots, x_n \in V$, where n is the input sequence length. The initial latents $h_1^{(0)}, \ldots, h_n^{(0)} \in \mathbb{R}^d$ are obtained from a learned embedding dictionary that contains one fixed vector per vocabulary token. Each of these latents is incrementally updated layer by layer by adding a residual. The residual added to the latent at position i in layer j is a function f_j of all preceding tokens' latents $h_1^{(j-1)}, \ldots, h_i^{(j-1)}$:

$$h_i^{(j)} = h_i^{(j-1)} + f_j\left(h_1^{(j-1)}, \ldots, h_i^{(j-1)}\right), \quad (1)$$

where the resulting vector $h_i^{(j)}$ is still of dimension d. The function f_j itself, called a transformer block, is composed of a masked self-attention layer followed by a feed-forward layer with a residual connection and root mean square (RMS) normalization in between (Vaswani et al., 2017; Touvron et al., 2023). Due to RMS normalization, all latents lie on a d-dimensional hypersphere of radius $\sqrt{d}$.

In pretraining, all transformer blocks $f_1, \ldots, f_m$ (with m the number of layers) are tuned such that the final latent $h_i^{(m)}$ for position i is well-suited for predicting the token at position $i+1$. For prediction, the final embedding vector is multiplied with a so-called *unembedding matrix* $U \in \mathbb{R}^{v \times d}$, which yields a real vector $z_i = U h_i^{(m)} \in \mathbb{R}^v$ containing a so-called *logit* score z_{it} for each vocabulary token $t \in V$. These scores are then transformed into probabilities $P(x_{i+1} = t \,|\, x_1, \ldots, x_i) \propto e^{z_{it}}$ via the softmax operation.

3.2 Interpreting latent embeddings: Logit lens

When transformers are deployed in practice, only the final latent vectors after the last transformer block are turned into token distributions by multiplying them with U and taking a softmax. However, since latents have the same shape in all layers, any latent can in principle be turned into a token distribution, by treating it as though it were a final-layer latent. Prematurely decoding tokens from latents this way, a method called the *logit lens* (cf. Sec. 2), can facilitate the inspection and interpretation of the internal state of transformers. Using the logit lens, we obtain one next-token distribution $P(x_{i+1} \,|\, h_i^{(j)})$ per position i and layer j.

We illustrate the logit lens in Fig. 1, where every cell shows the most likely next token when applying the logit lens to the latent in that position and layer. As seen, the logit lens decodes contextually appropriate tokens already in intermediate layers.

3.3 Data: Tasks for eliciting latent language

Our goal is to explore whether Llama-2's internal, latent states correspond to specific natural languages. Although the logit lens allows us to map latent vectors to token distributions, we still require a mapping from token distributions to languages.

Doing so in general is difficult as many tokens are ambiguous with respect to language; e.g., the token "an" is commonly used in English, French, and German, among others. To circumvent this issue, we construct prompts $x_1 \ldots x_n$ where the correct next token x_{n+1} is (1) obvious and (2) can be unambiguously attributed to one language.

Prompt design. To ensure that the next token is obvious (criterion 1), we design three text completion tasks where the next token x_{n+1} can be easily inferred from the prompt $x_1 \ldots x_n$. In describing the tasks, we use Chinese as an example language.

Translation task. Here the task is to translate the preceding non-English (e.g., French) word to Chinese. We show the model four words with their correct translations, followed by a fifth word without its translation, and let the model predict the next token ("中文" means "Chinese" below):

```
Français: "vertu" - 中文: "德"
Français: "siège" - 中文: "座"
Français: "neige" - 中文: "雪"
Français: "montagne" - 中文: "山"
Français: "fleur" - 中文: "
```

With such a prompt, Llama-2 can readily infer that it should translate the fifth French word. We carefully select words as described below and construct one prompt per word by randomly sampling demonstrations from the remaining words.

Repetition task. Similarly, we task the model to simply repeat the last word, instead of translating it, by prompting as follows:

```
中文: "德" - 中文: "德"
中文: "座" - 中文: "座"
中文: "雪" - 中文: "雪"
中文: "山" - 中文: "山"
中文: "花" - 中文: "
```

Cloze task. As a slightly harder task, we consider a cloze test, where the model must predict a masked word in a sentence. Given a target word, we construct an English sentence starting with the word by prompting GPT-4, mask the target word, and translate the sentence to the other languages. To construct prompts, we sample two demonstrations

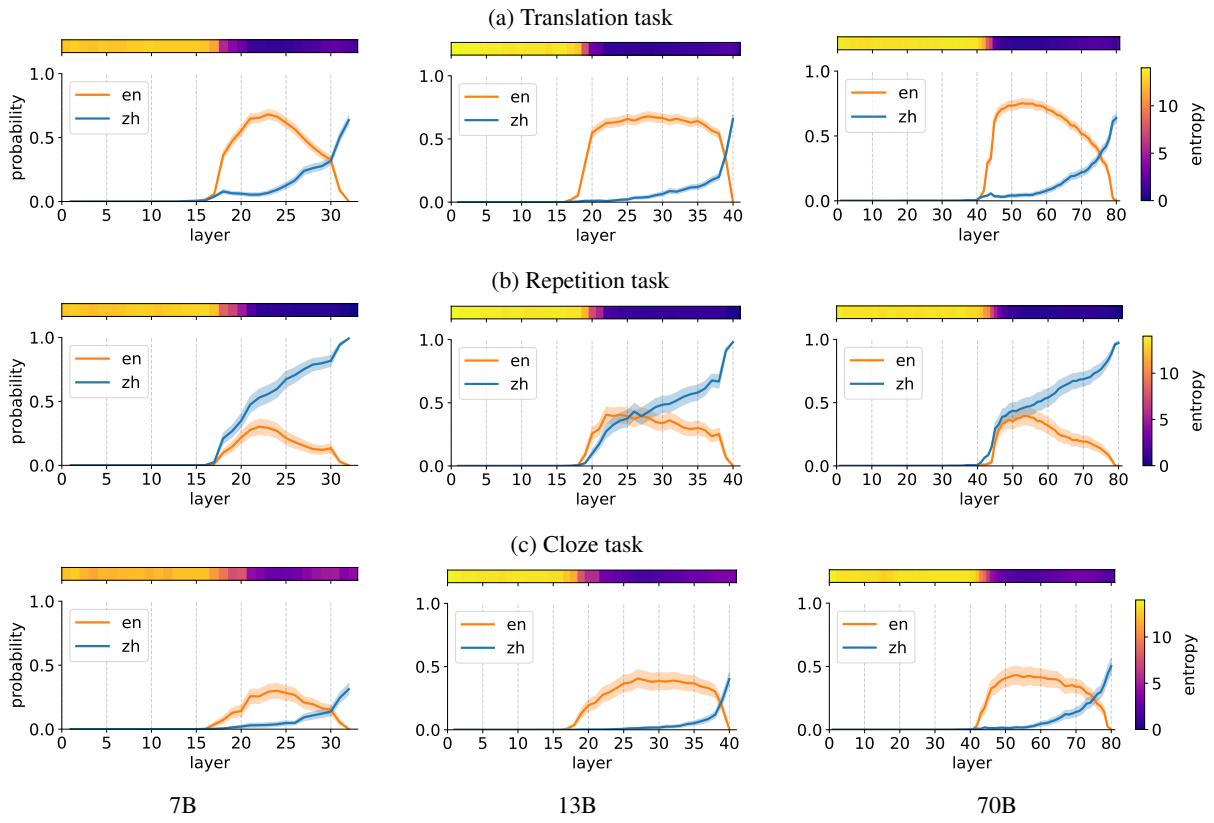

Figure 2: **Language probabilities for latents during Llama-2 forward pass,** for **(a)** translation task from union of German/French/Russian to Chinese, **(b)** Chinese repetition task, **(c)** Chinese cloze task. Each task evaluated for model sizes (columns) 7B, 13B, 70B. On x-axes, layer index; on y-axes, probability (according to logit lens) of correct Chinese next token (blue) or English analog (orange). Error bars show 95% Gaussian confidence intervals over input texts (353 for translation, 139 for repetition and cloze).

from the remaining words. An English example before translation to the other languages follows:

> A "___" is used to play sports like soccer and basketball. Answer: "ball".
> A "___" is a solid mineral material forming part of the surface of the earth. Answer: "rock".
> A "___" is often given as a gift and can be found in gardens. Answer: "

Word selection. To enable unambiguous language attribution (criterion 2), we construct a closed set of words per language. As a particularly clean case, we focus on Chinese, which has many single-token words and does not use spaces. We scan Llama-2's vocabulary for single-token Chinese words (mostly nouns) that have a single-token English translation. This way, Llama-2's probabilities for the correct next Chinese word and for its English analog can be directly read off the next-token probabilities.

For robustness, we also run all experiments on German, French, and Russian. For this, we translate the selected Chinese/English words and, for each language, discard words that share a token prefix with the English version, as this would render language detection (cf. Sec. 3.4) ambiguous.

We work with 139 Chinese, 104 German, 56 French, and 115 Russian words (cf. Appendix A.1).

3.4 Measuring latent language probabilities

To investigate a hypothetical pivot language inside Llama-2, we apply the logit lens to the latents $h_n^{(j)}$ corresponding to the last input token x_n for each layer j, obtaining one next-token distribution $P(x_{n+1} \mid h_n^{(j)})$ per layer. Our prompts (cf. Sec. 3.3) are specifically designed such that an intermediate next-*token* distribution lets us estimate the probability of the correct next *word* in the input language as well as English. Since we specifically select single-token words in Chinese (ZH) as well as English (EN), we can simply define the probability of language $\ell \in \{\text{ZH}, \text{EN}\}$ as the probability of the next token being ℓ's version t_ℓ of the correct single-token word: $P(\text{lang} = \ell \mid h_n^{(j)}) := P(x_{n+1} = t_\ell \mid h_n^{(j)})$. (For readability we also simply write $P(\text{lang} = \ell)$.)

Note that this does not define a distribution over languages, as generally $\sum_\ell P(\text{lang} = \ell) < 1$.

In other languages (and in corner cases in Chinese and English), we must account for multiple tokenizations and whitespaces (cf. Appendix A.2).

4 Results

When presenting results, we first (Sec. 4.1) take a probabilistic view via the logit lens (Sec. 3.2), for all tasks and all model sizes. (Since the results are consistent across languages, we focus on Chinese here and refer to Appendix B for French, German, and Russian.) Then (Sec. 4.2) we drill deeper by taking a geometric view of how token embeddings drift as the transformer computes layer by layer.

4.1 Probabilistic view: Logit lens

The logit lens gives us one set of language probabilities (cf. Sec. 3.4) per input prompt and layer. Fig. 2 tracks the evolution of language probabilities from layer to layer, with one plot per combination of model size (columns) and task[1] (rows). The x-axes show layer indices, and the y-axis the language probabilities $P(\text{lang} = \text{ZH})$ and $P(\text{lang} = \text{EN})$ averaged over input prompts.

On the translation and cloze tasks a consistent picture emerges across model sizes. Neither the correct Chinese token nor its English analog garner any noticeable probability mass during the first half of layers. Then, around the middle layer, English begins a sharp rise followed by a decline, while Chinese slowly grows and, after a crossover with English, spikes on the last five layers. On the repetition task, Chinese already rises alongside English (discussed in Sec. 6). This is in contrast to all other languages, where English rises first (Appendix B).

On top of the language probabilities (Sec. 3.4), the entropy of the full next-token distribution is shown as a heatmap above the plots. We again observe a consistent pattern across tasks and model sizes: high entropy in the first half of layers, while both $P(\text{lang} = \text{ZH})$ and $P(\text{lang} = \text{EN})$ are close to zero, followed by a sharp drop at the same time that $P(\text{lang} = \text{EN})$ rises. From there on, entropy remains low, with a slight rebound as probability mass shifts from English to Chinese.

With $32{,}000 \approx 2^{15}$ tokens in the vocabulary, the early entropy of around 14 bits implies a close-to-uniform next-token distribution (around 15 bits).

[1] In Fig. 2, translation task uses union of German, French, and Russian as source languages. For individual source languages, as well as all target languages, cf. Appendix B.

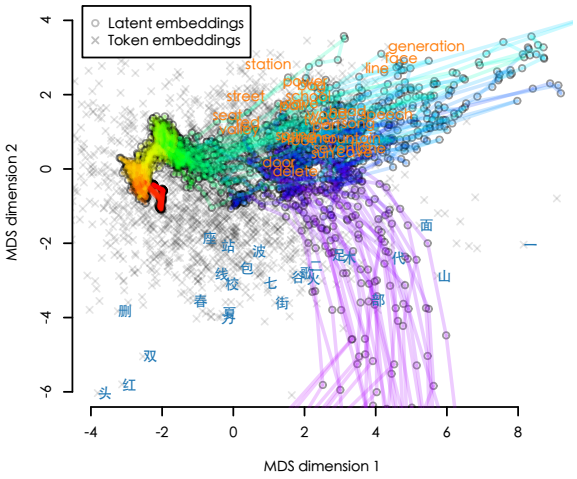

Figure 3: **Latent trajectories through transformer layers.** 2D embedding of latents (○) and output tokens (×) found via multidimensional scaling. Latents for same prompt connected by rainbow-colored path, proceeding from layer 1 (red) to 80 (violet). Labels for correct Chinese next tokens (one per prompt) in blue, for English analogs in orange. Takeaway: latents reach correct Chinese token after detour through English.

Path visualization. The plots of Fig. 2 only consider the probability of the correct Chinese next token and its English analog, without speaking to the remaining tokens. To form an intuition of the entire distribution, we use dimensionality reduction to visualize the data. First, we define the distance between a latent h_n at position n and a token t via the negative log-likelihood of t given h_n, as computed by the logit lens (cf. Sec. 3.4): $d(h_n, t) = -\log P(x_{n+1} = t \,|\, h_n)$. Then, we use classical multidimensional scaling to embed tokens and latents in an approximately distance-preserving joint 2D space. (Intra-token and intra-latent distances are set to $\max_{h,t} d(h,t)$, which serves as a "spring force" pushing the 2D points apart.)

A transformer's forward computation for a given final input token x_n can now be visualized by connecting the 2D embeddings of the latents $h_n^{(j)}$ in subsequent layers j, as presented and explained in Fig. 3 (German-to-Chinese translation, 70B). We make two observations: (1) An English and a Chinese token cluster emerges, suggesting that the same latent also gives high probability to an entire language, in addition to the language-specific version of the correct next token. (2) Paths first pass through the English cluster, and only later reach the Chinese cluster. Taken together, the emerging picture is that, when translating a German word