Deterministic indexing for packed strings

Given a string $S$ of length $n$, the classic string indexing problem is to preprocess $S$ into a compact data structure that supports efficient subsequent pattern queries. In the deterministic variant the goal is to solve the string indexing problem without any randomization (at preprocessing time or query time). In the packed variant the strings are stored with several character in a single word, giving us the opportunity to read multiple characters simultaneously. Our main result is a new string index in the deterministic and packed setting. Given a packed string $S$ of length $n$ over an alphabet $\sigma$, we show how to preprocess $S$ in $O(n)$ (deterministic) time and space $O(n)$ such that given a packed pattern string of length $m$ we can support queries in (deterministic) time $O\left(\frac{m}{\alpha} + \log m + \log \log \sigma\right)$, where $\alpha = \frac{w}{\log \sigma}$ is the number of characters packed in a word of size $w = \theta(\log n)$. Our query time is always at least as good as the previous best known bounds and whenever several characters are packed in a word, i.e., $\log \sigma$