```python
# chunk_text_dynamic

import pytest
from agentparse import chunk_text_dynamic


# Test normal case with a standard text input
def test_chunk_text_dynamic_normal_case():
    text = "This is a simple test to check the chunking functionality of the text."
    chunks = chunk_text_dynamic(text, limit_tokens=10)
    assert len(chunks) == 1
    assert (
        chunks[0]
        == "This is a simple test to check the chunking functionality of the text."
    )


# Test chunking with a limit that requires multiple chunks
def test_chunk_text_dynamic_multiple_chunks():
    text = (
        "This is a long text that should be split into multiple chunks based on the token limit. "
        * 5
    )
    chunks = chunk_text_dynamic(text, limit_tokens=15)
    assert len(chunks) > 1
    assert all(len(chunk.split()) <= 15 for chunk in chunks)
```

```python
# Test edge case with an empty string
def test_chunk_text_dynamic_empty_string():
    text = ""
    chunks = chunk_text_dynamic(text, limit_tokens=10)
    assert chunks == []


# Test edge case with a single word exceeding the limit
def test_chunk_text_dynamic_single_word_exceeding_limit():
    text = "supercalifragilisticexpialidocious"
    chunks = chunk_text_dynamic(text, limit_tokens=5)
    assert len(chunks) == 1
    assert chunks[0] == "supercalifragilisticexpialidocious"


# Test edge case with a limit of zero
def test_chunk_text_dynamic_zero_limit():
    text = "This text should not be chunked."
    with pytest.raises(
        ValueError, match="Limit must be greater than zero"
    ):
        chunk_text_dynamic(text, limit_tokens=0)
```

```
# Test chunking with punctuation
def test_chunk_text_dynamic_with_punctuation():
    text = "Hello, world! This is a test. Let's see how it handles punctuation."
    chunks = chunk_text_dynamic(text, limit_tokens=10)
    assert len(chunks) == 1
    assert "Hello, world!" in chunks[0]
    assert "Let's see how it handles punctuation." in chunks[0]
```