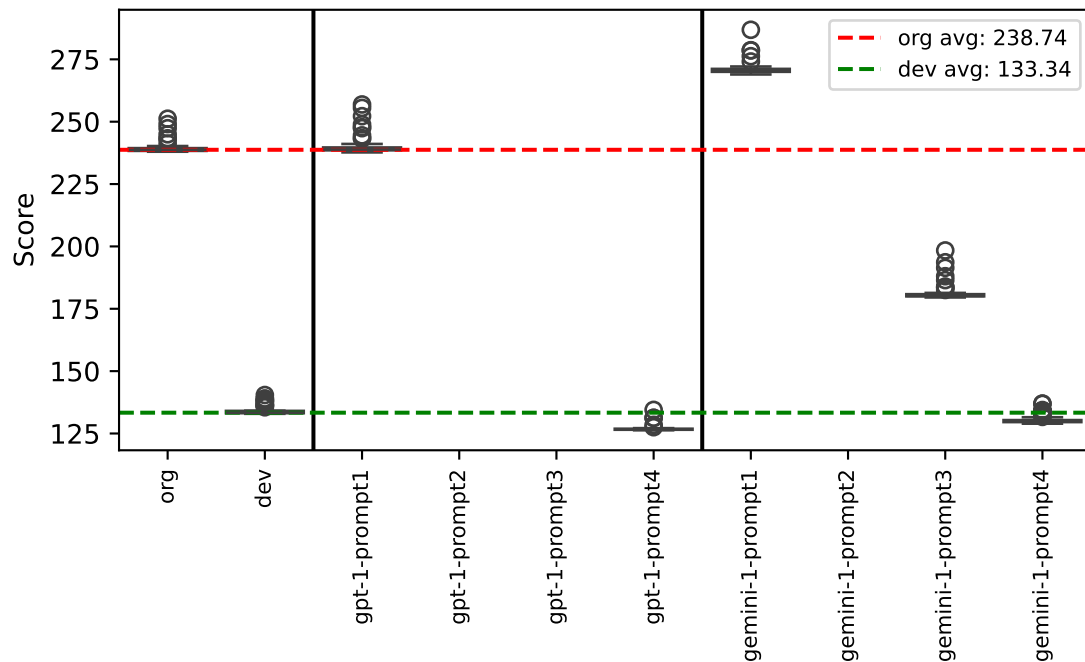


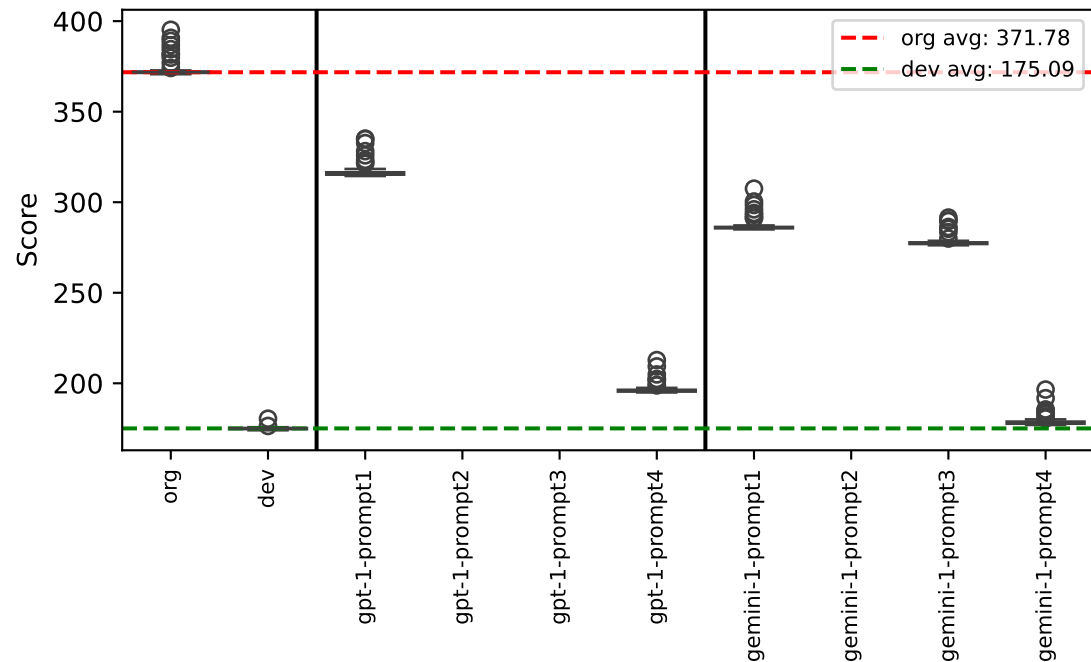
Project: RoaringBitmap

Benchmark: org.roaringbitmap.CheckedAddBenchmark.optimized

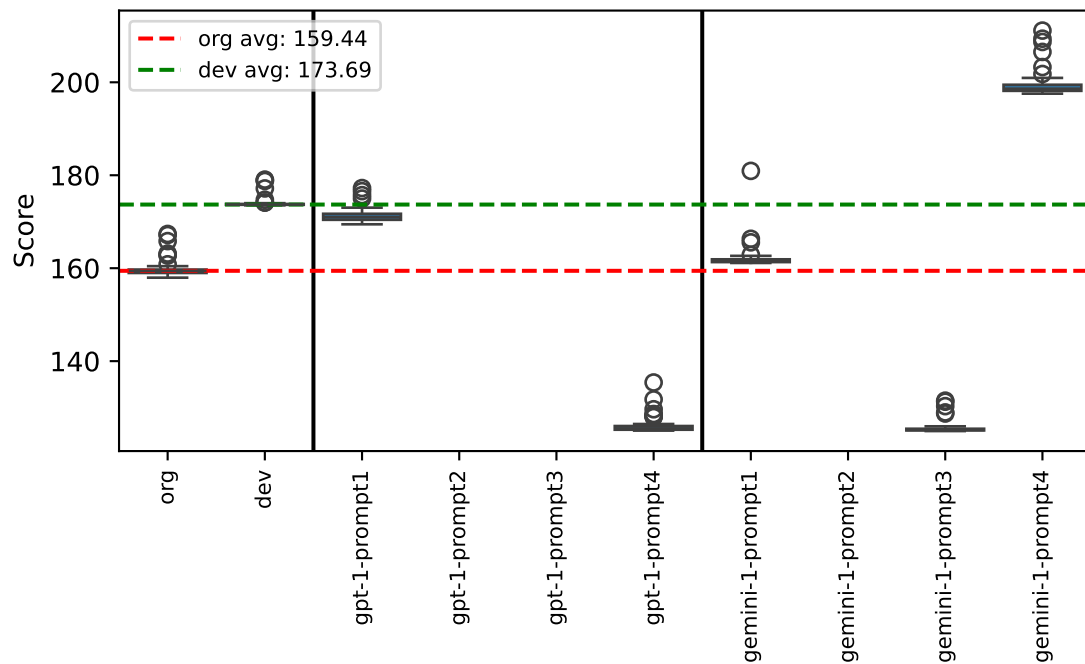
Params: {'containerType': 'RUN', 'inputData': 'HALF_DUPLICATED'}



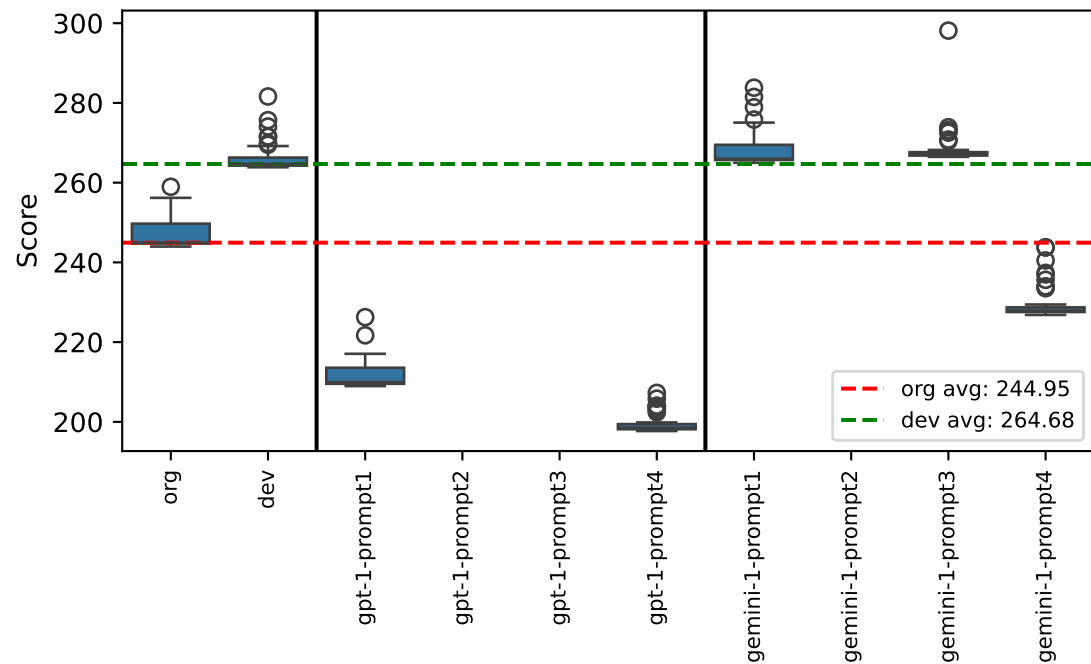
Params: {'containerType': 'RUN', 'inputData': 'UNIQUE'}



Params: {'containerType': 'BITMAP', 'inputData': 'HALF_DUPLICATED'}

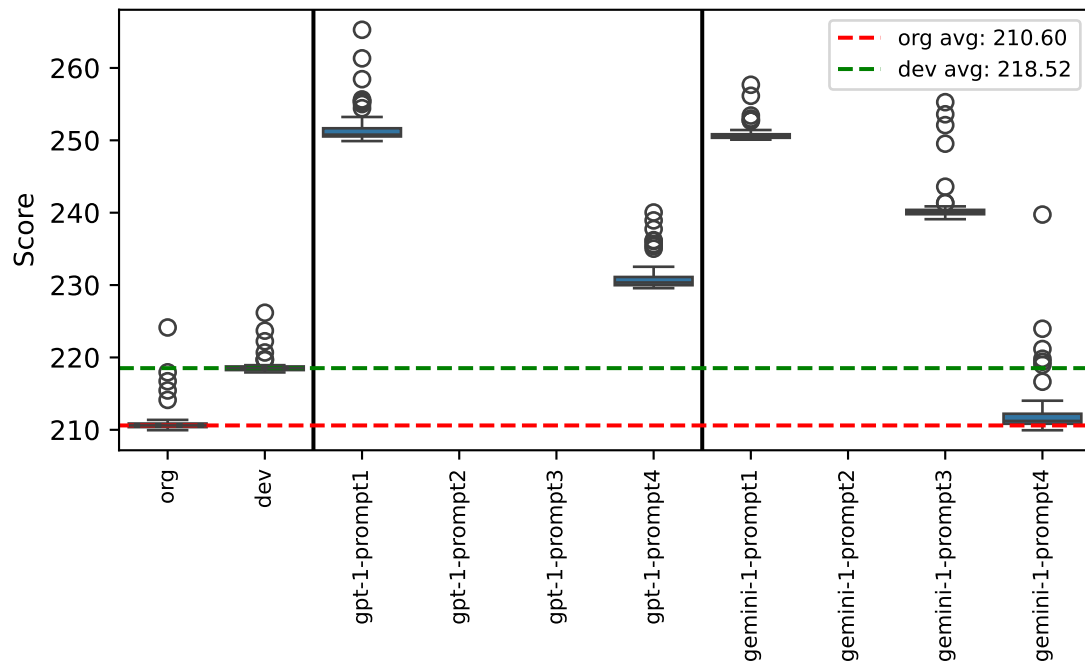


Params: {'containerType': 'BITMAP', 'inputData': 'UNIQUE'}

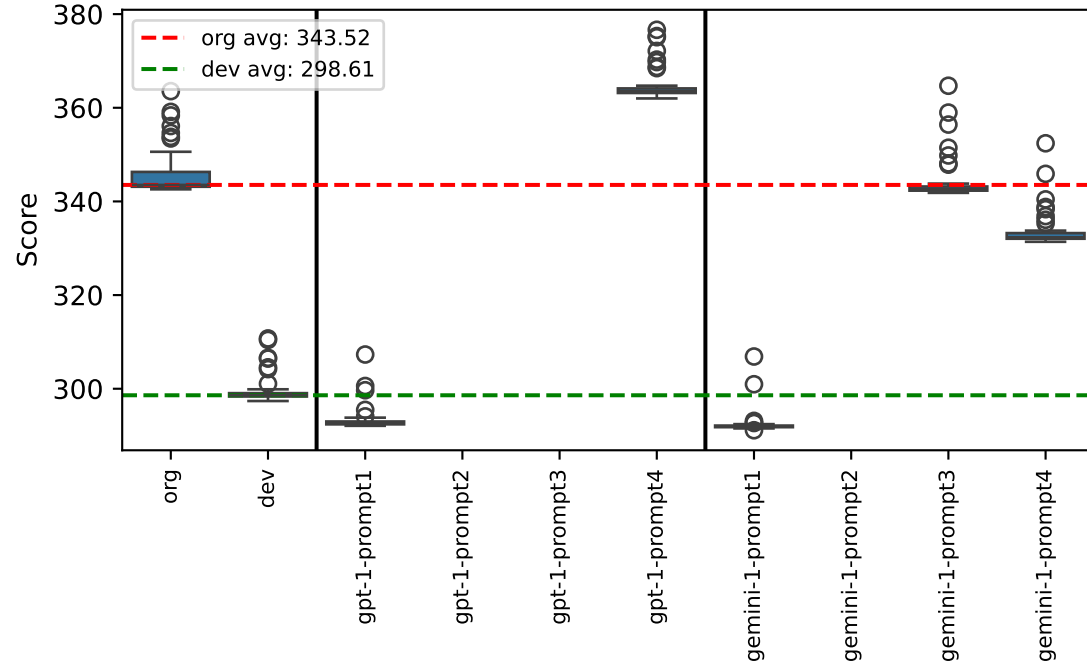


Benchmark: org.roaringbitmap.CheckedAddBenchmark.original

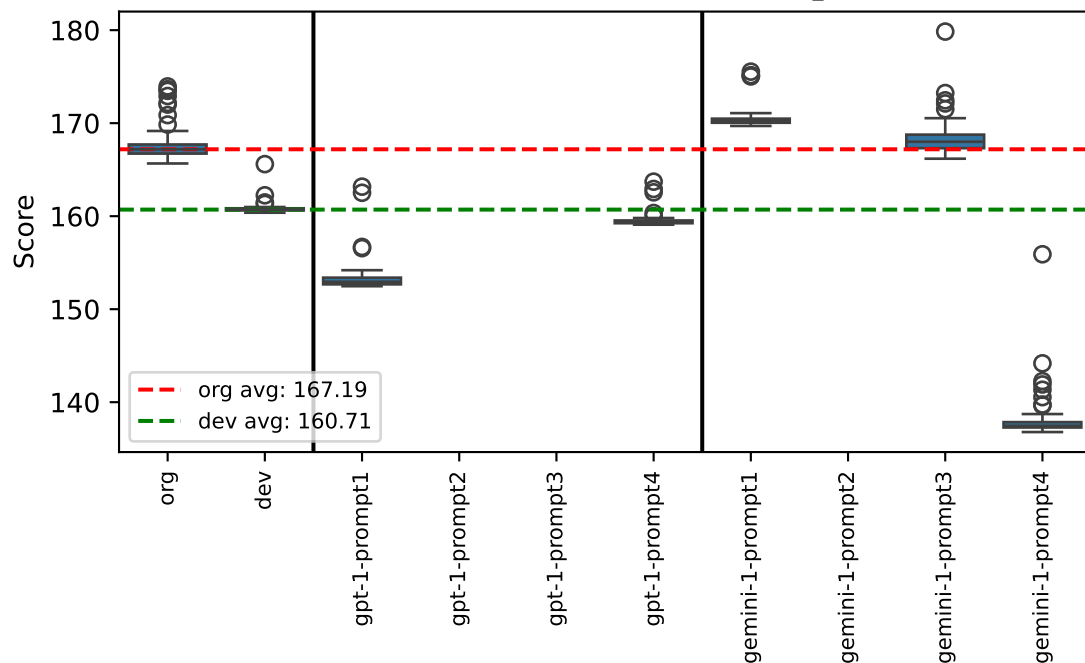
Params: {'containerType': 'RUN', 'inputData': 'HALF_DUPLICATED'}



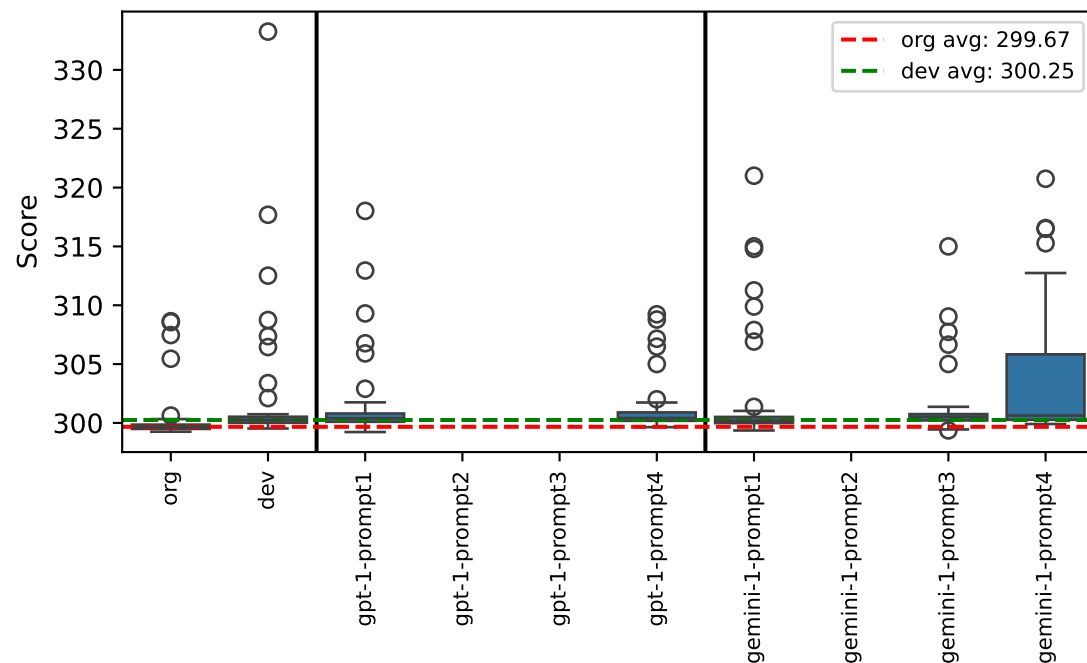
Params: {'containerType': 'RUN', 'inputData': 'UNIQUE'}



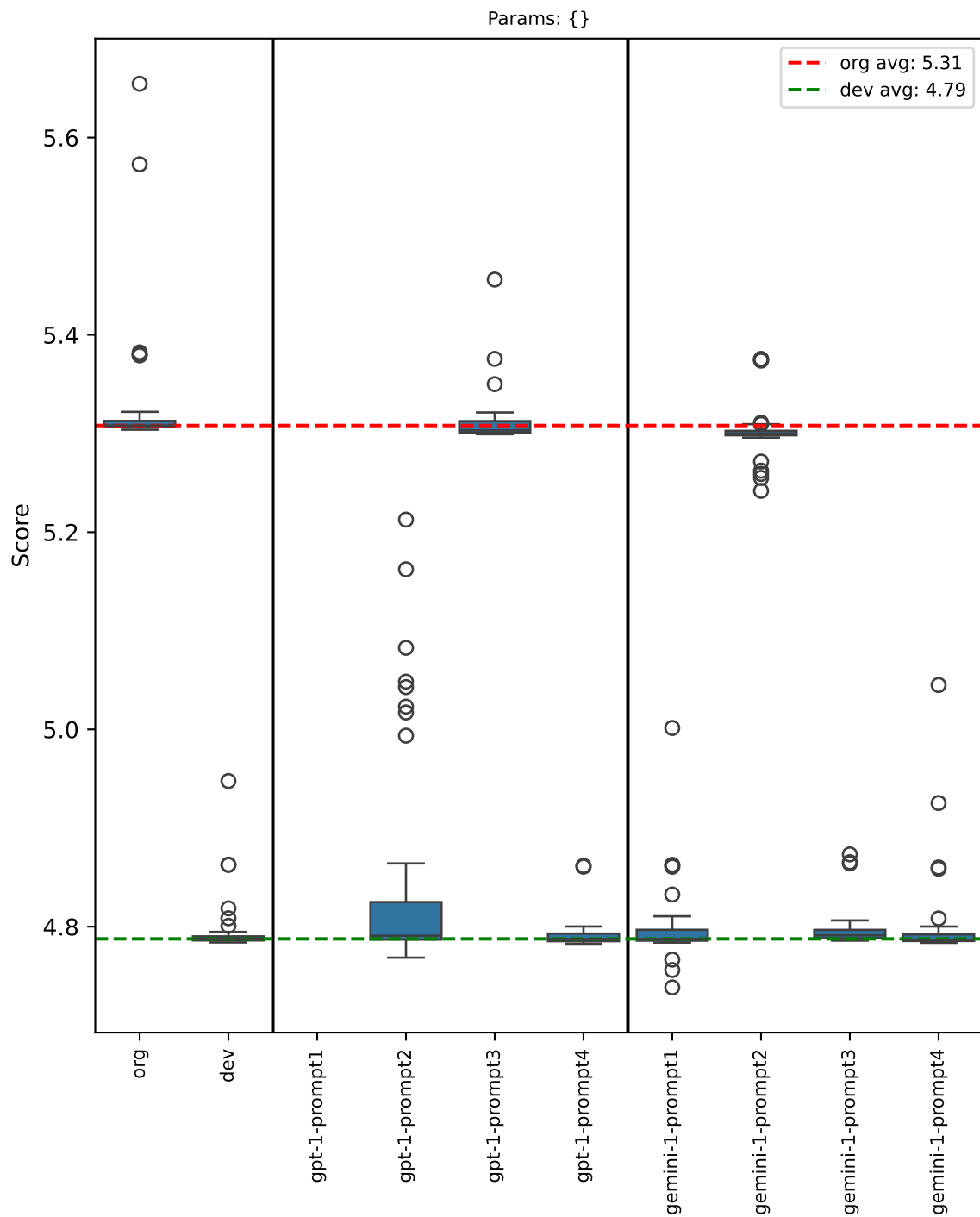
Params: {'containerType': 'BITMAP', 'inputData': 'HALF_DUPLICATED'}



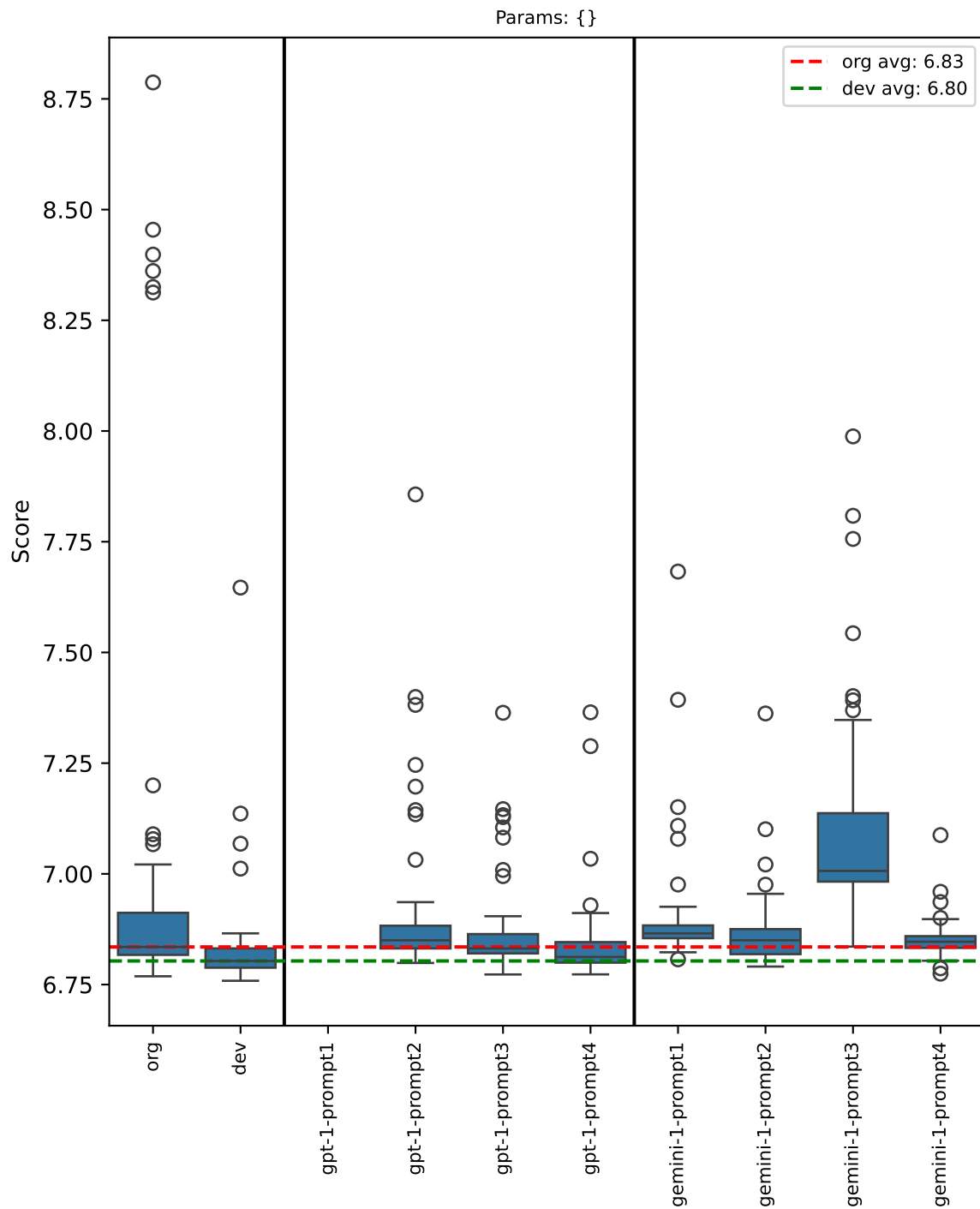
Params: {'containerType': 'BITMAP', 'inputData': 'UNIQUE'}



Benchmark: org.roaringbitmap.IntermediateByteArrayBenchmark.optimized

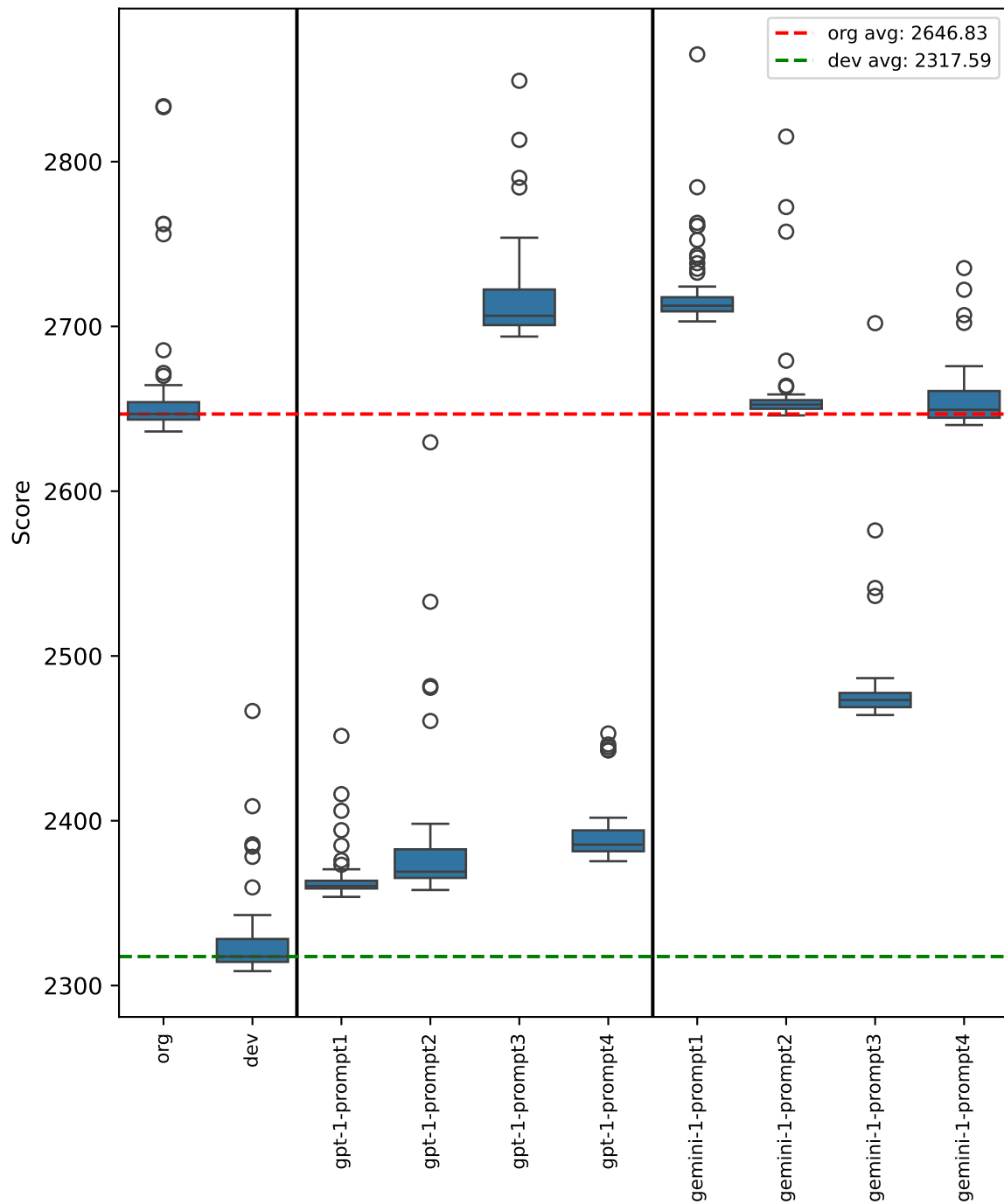


Benchmark: org.roaringbitmap.IntermediateByteArrayBenchmark.original



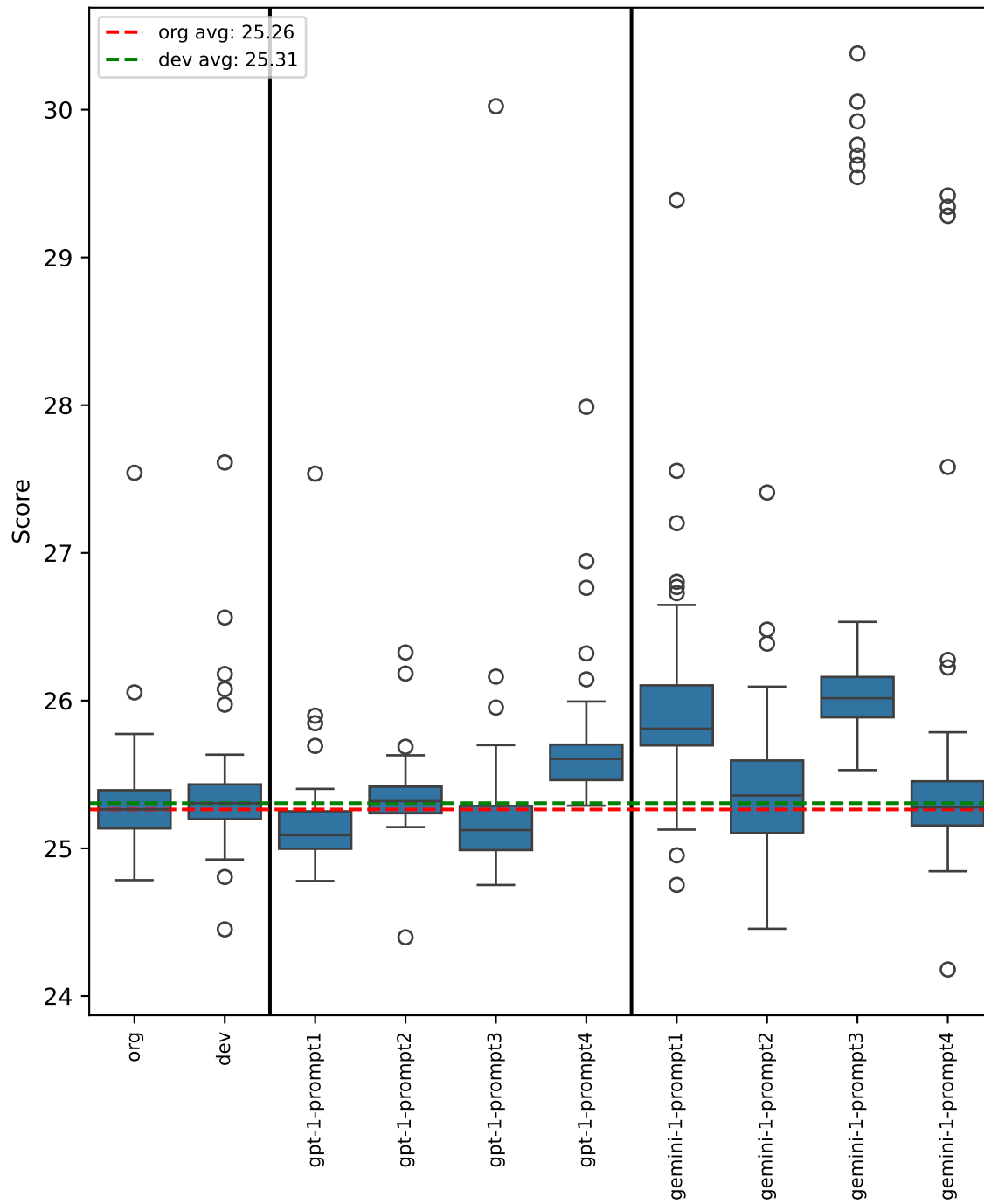
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.add

Params: {'count': '10000', 'n': '200'}



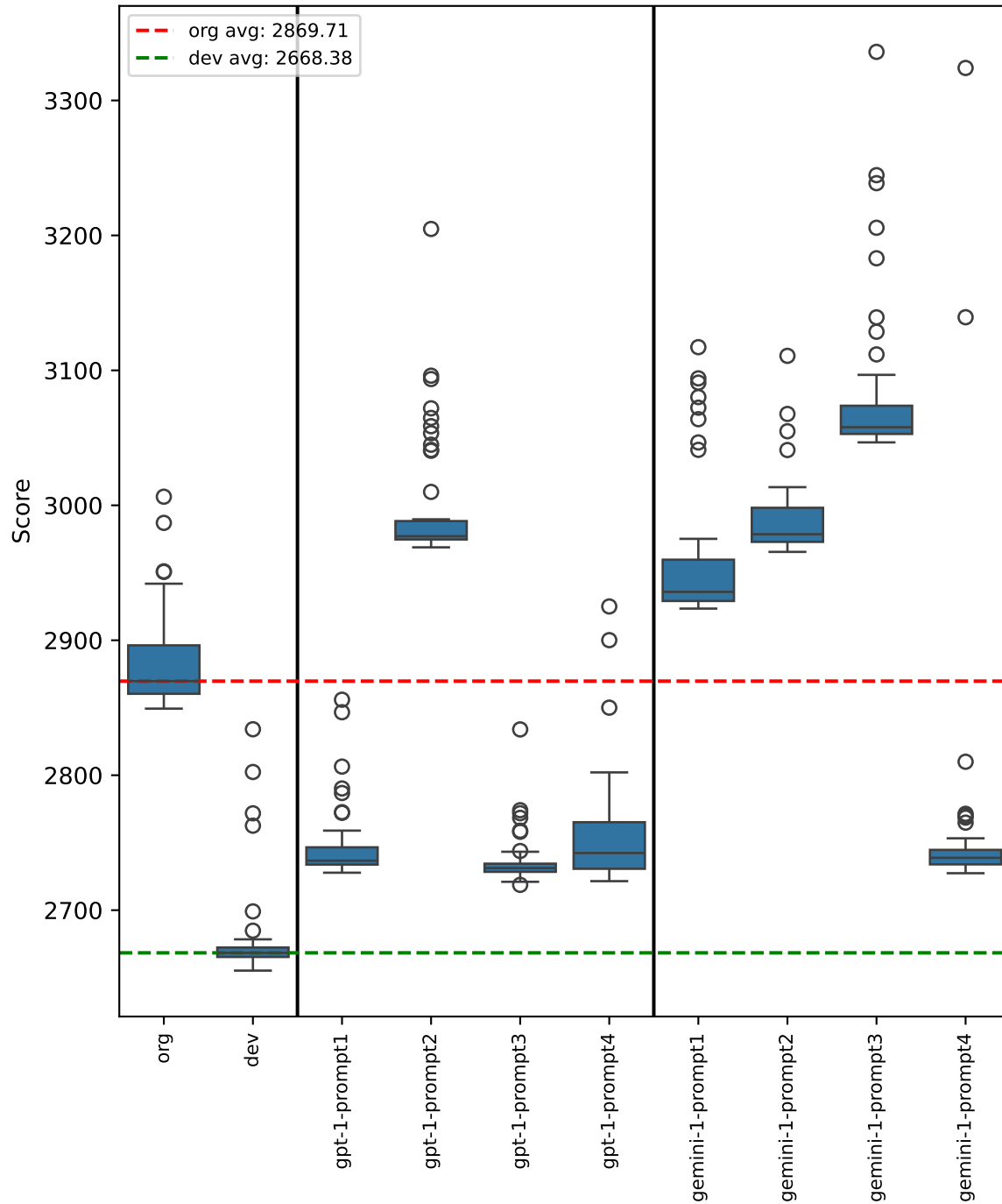
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.addIncludingAndNot

Params: {'count': '10000', 'n': '200'}



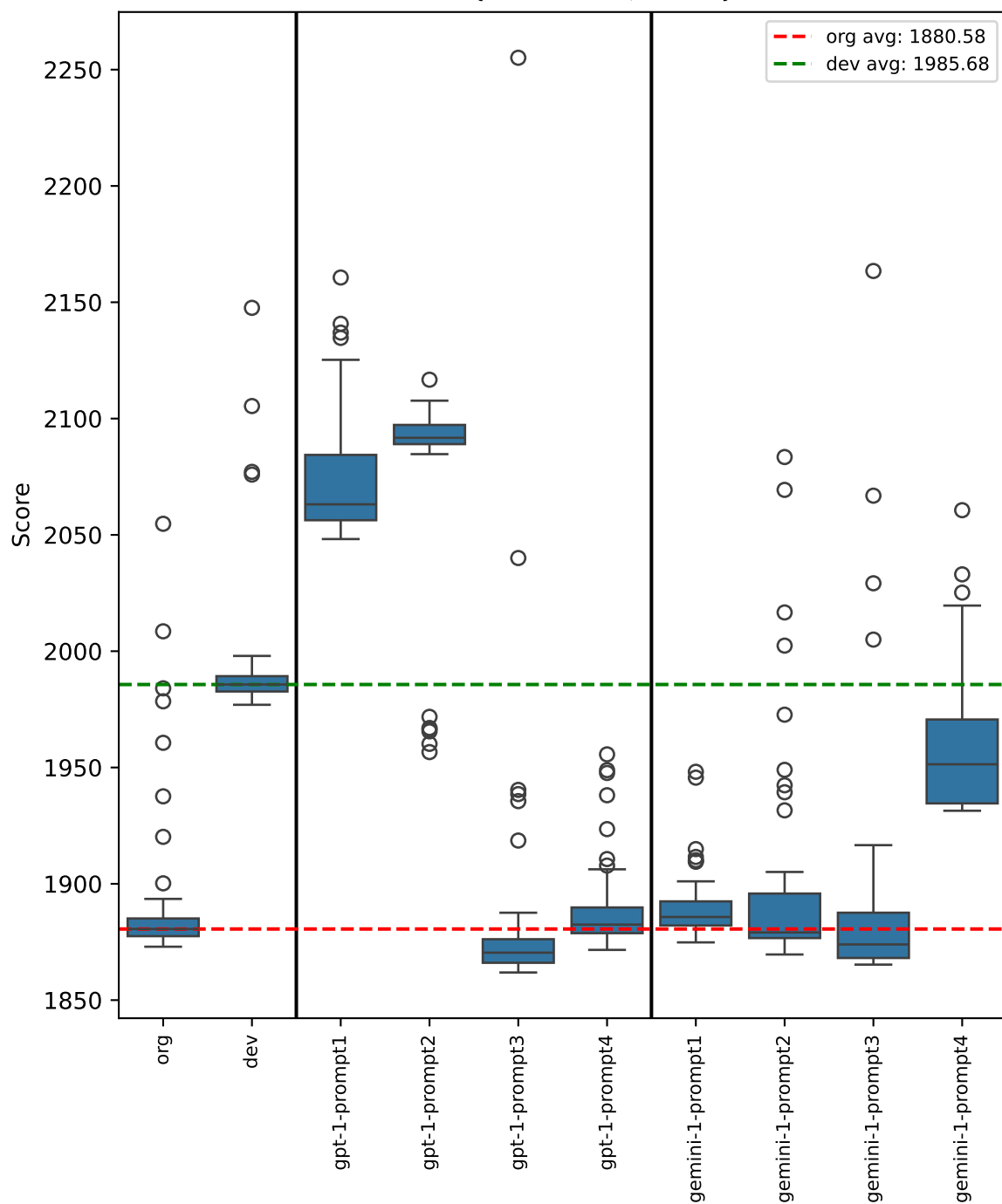
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.batchIterator

Params: {'count': '10000', 'n': '200'}

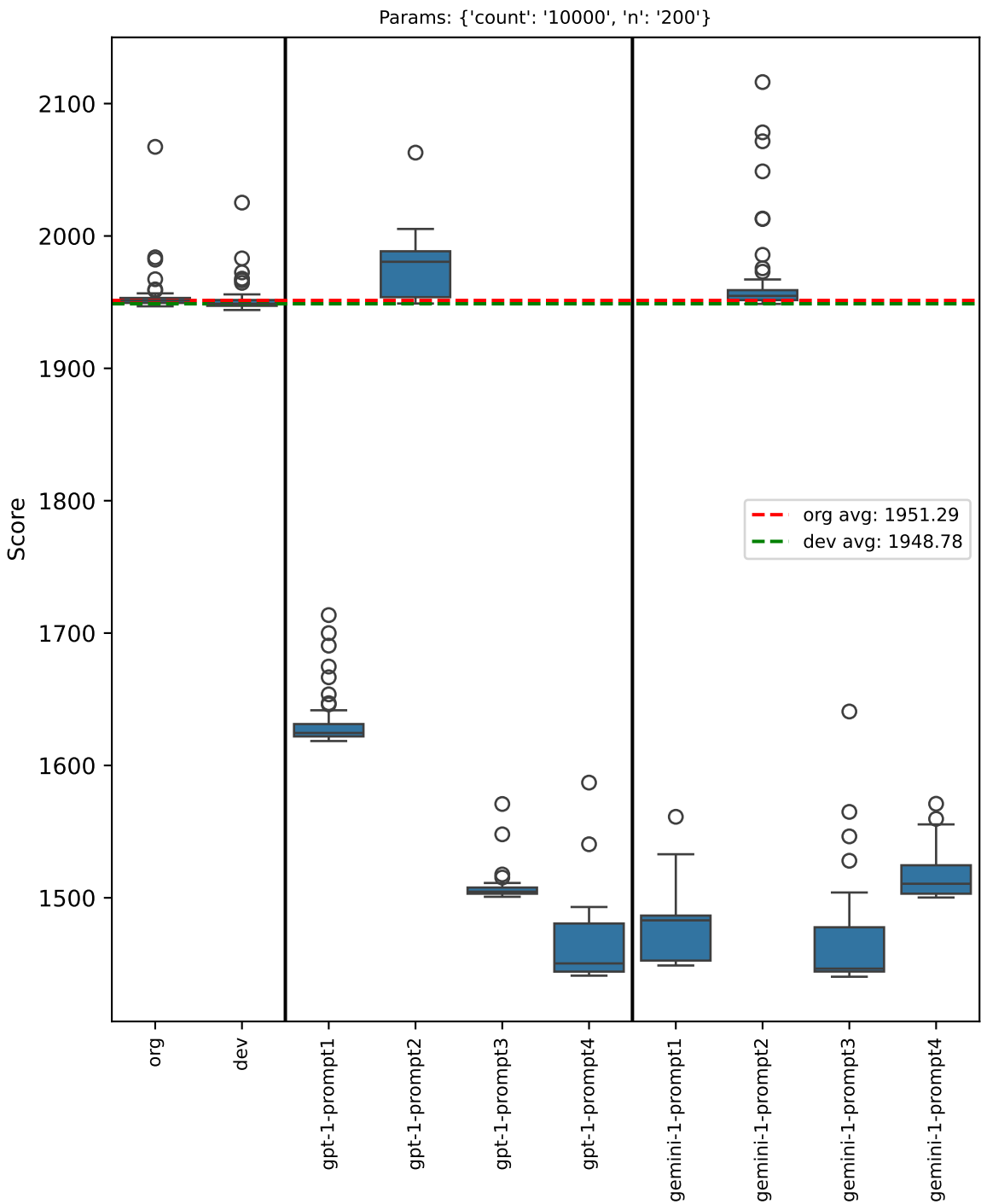


Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.batchIteratorAddAtOnce

Params: {'count': '10000', 'n': '200'}

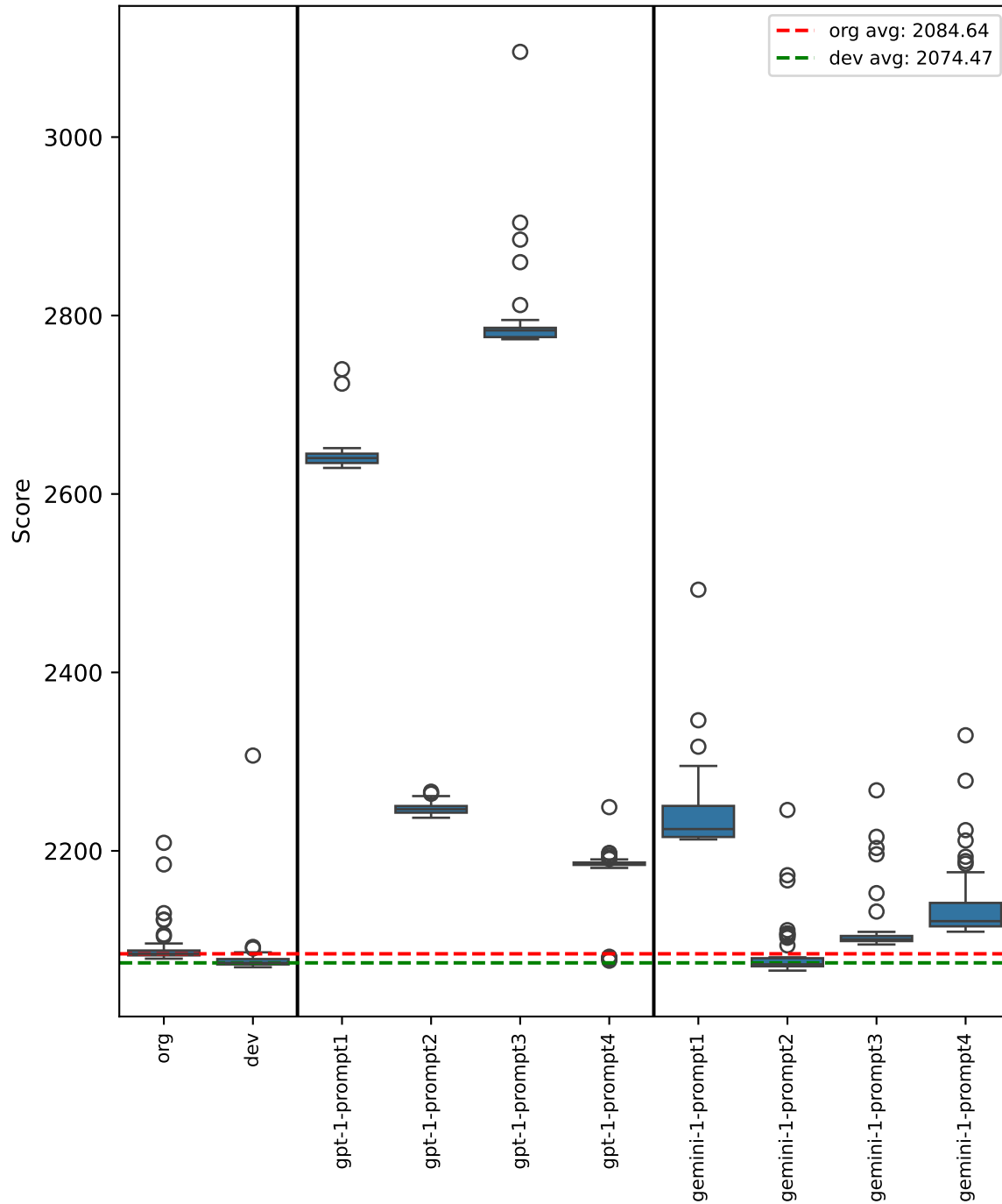


Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.batchIteratorAddAtOnceIncludingAndNot



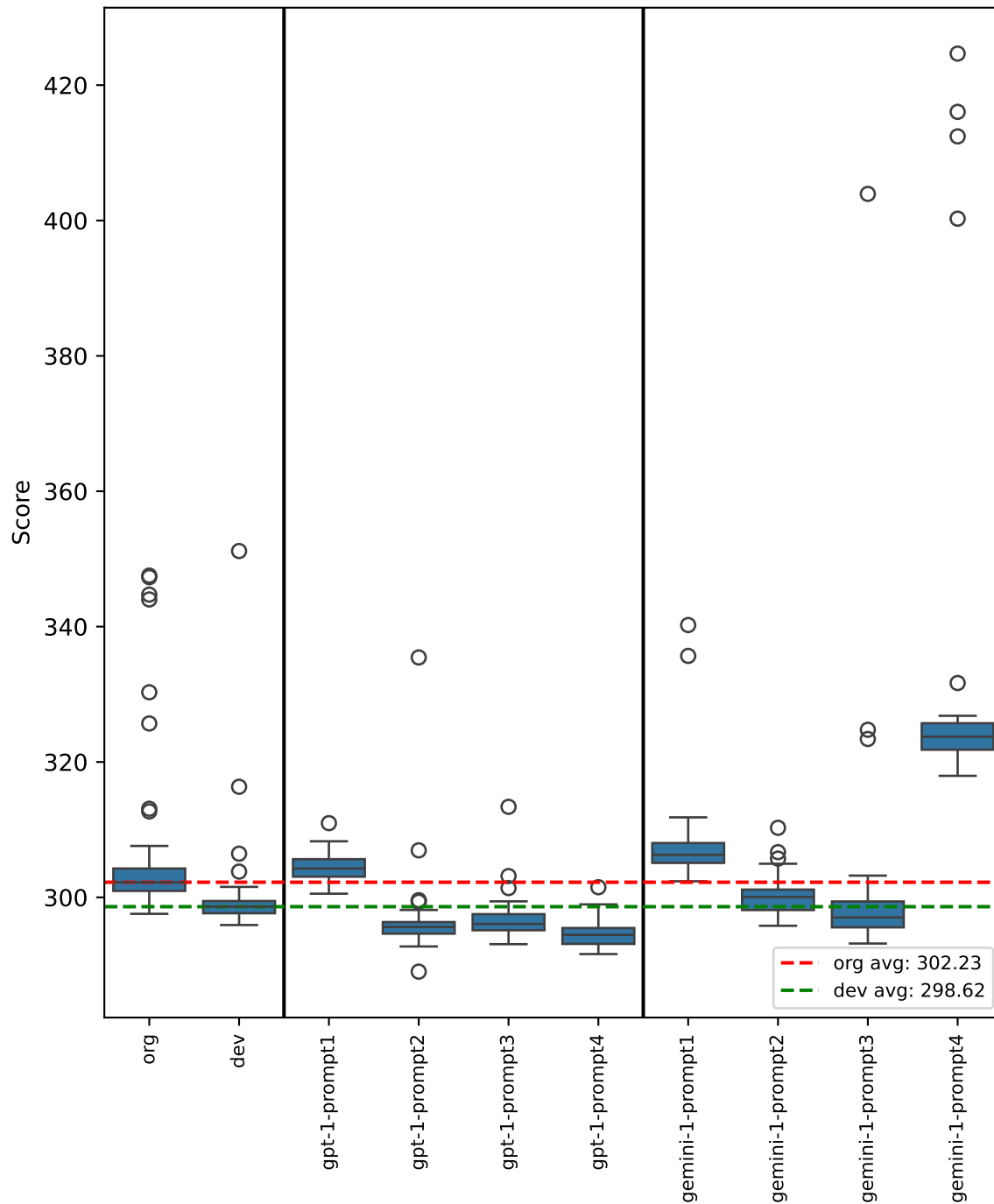
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.batchIteratorIncludingAndNot

Params: {'count': '10000', 'n': '200'}



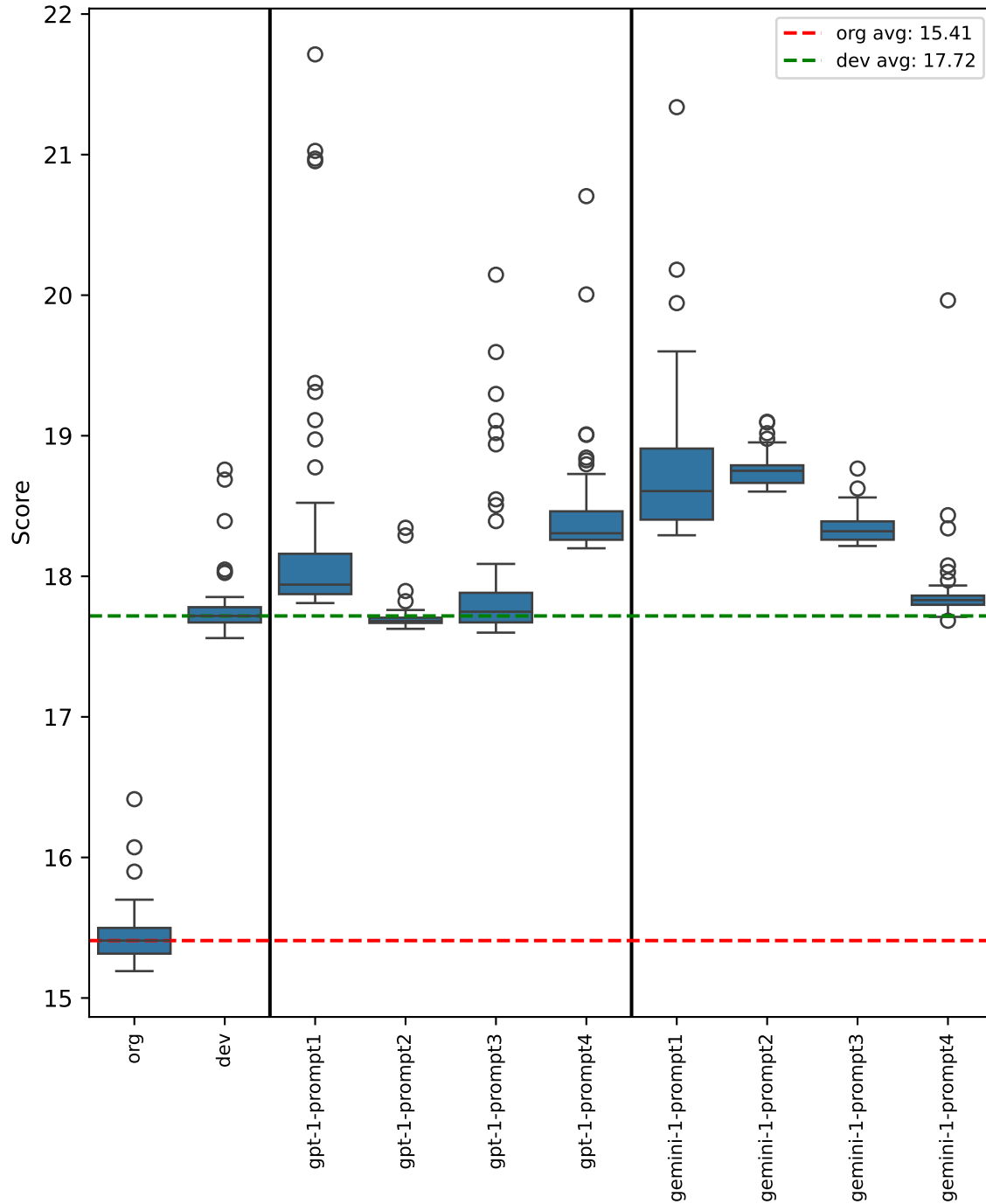
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.limit

Params: {'count': '10000', 'n': '200'}



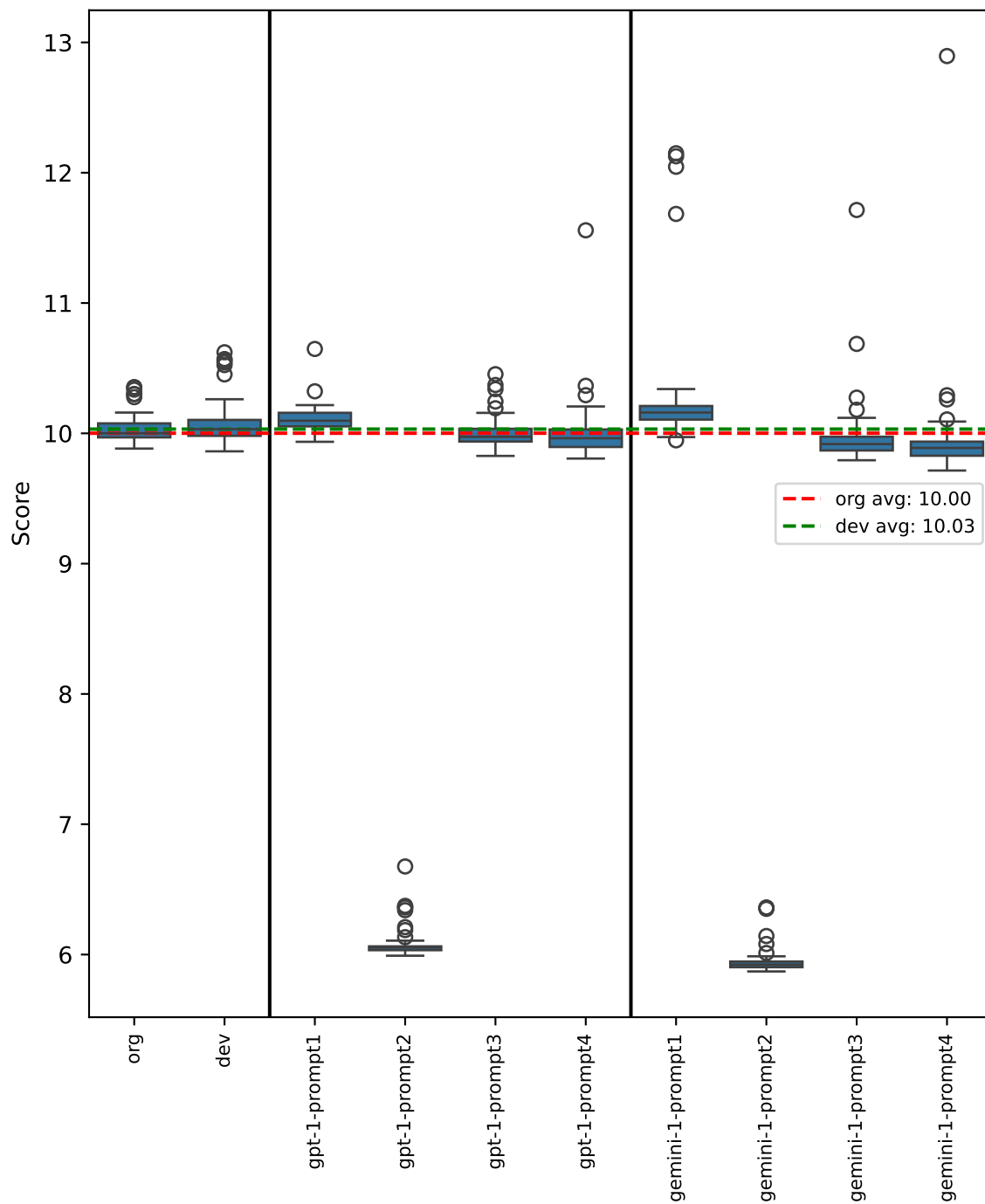
Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.limitIncludingAndNot

Params: {'count': '10000', 'n': '200'}

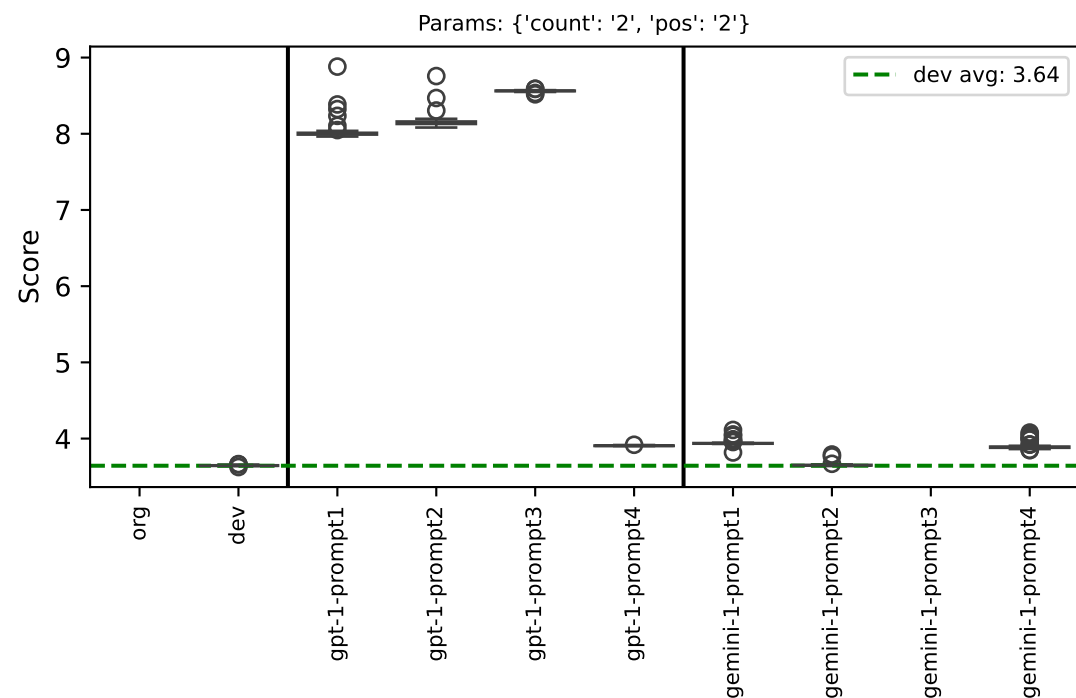
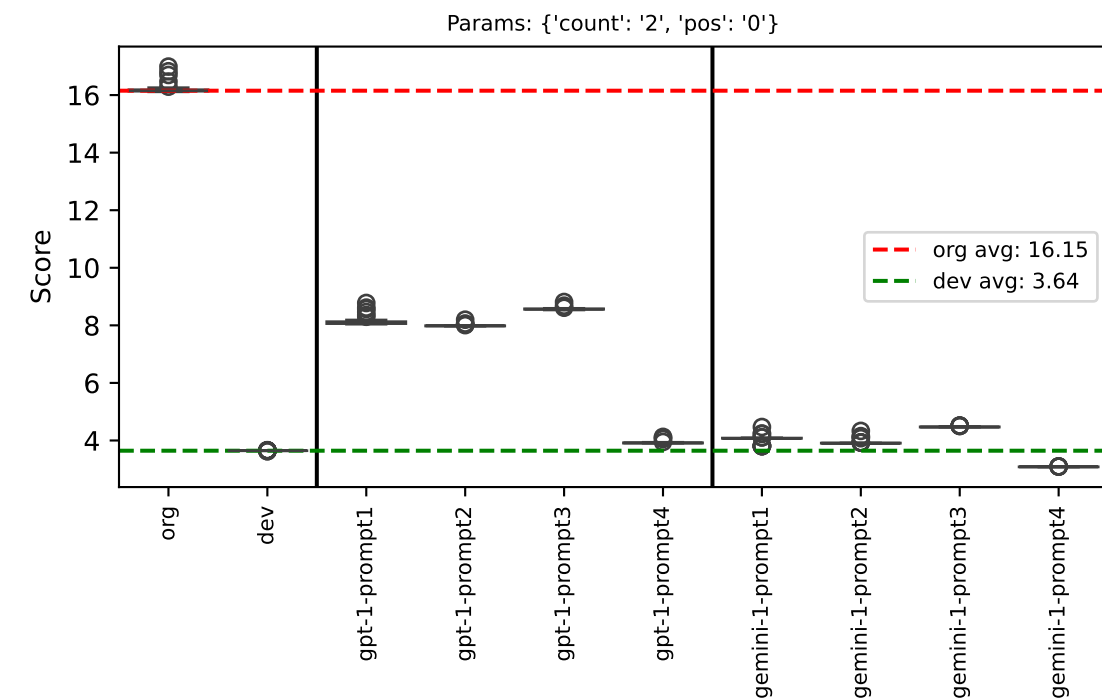
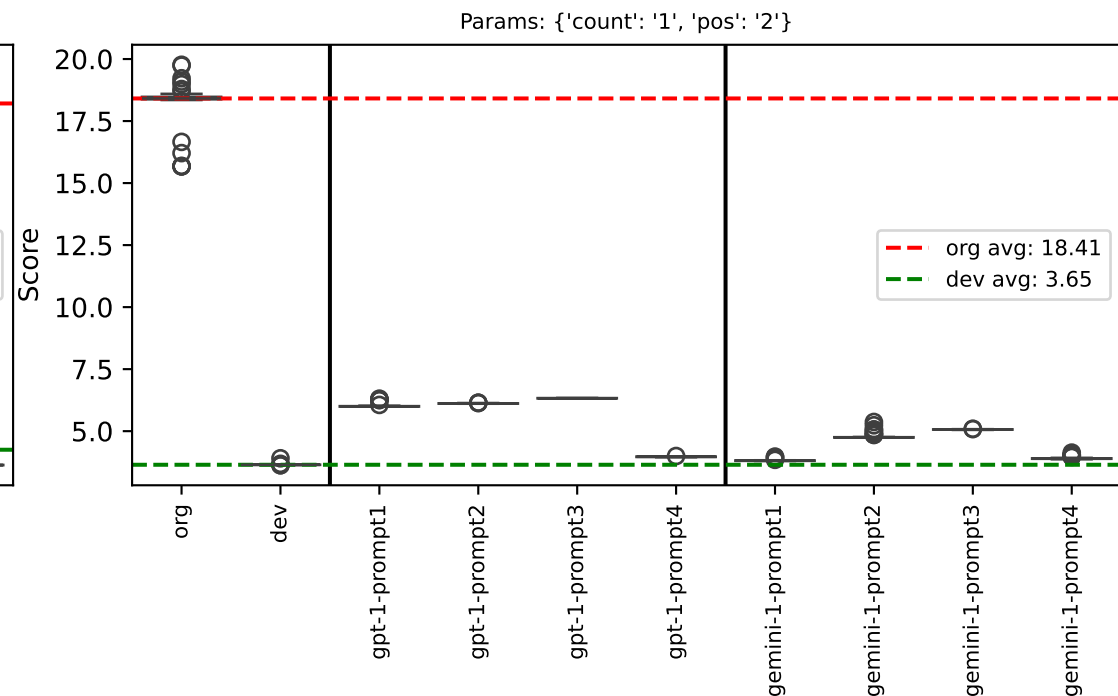
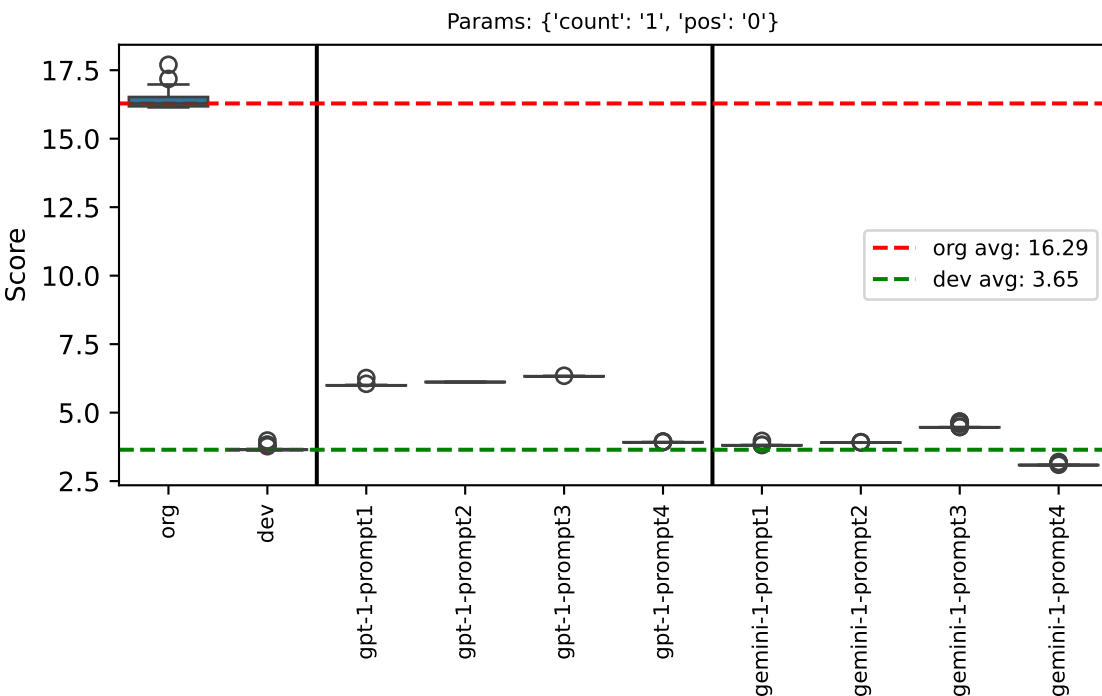


Benchmark: org.roaringbitmap.SelectTopValuesBenchmark.remove

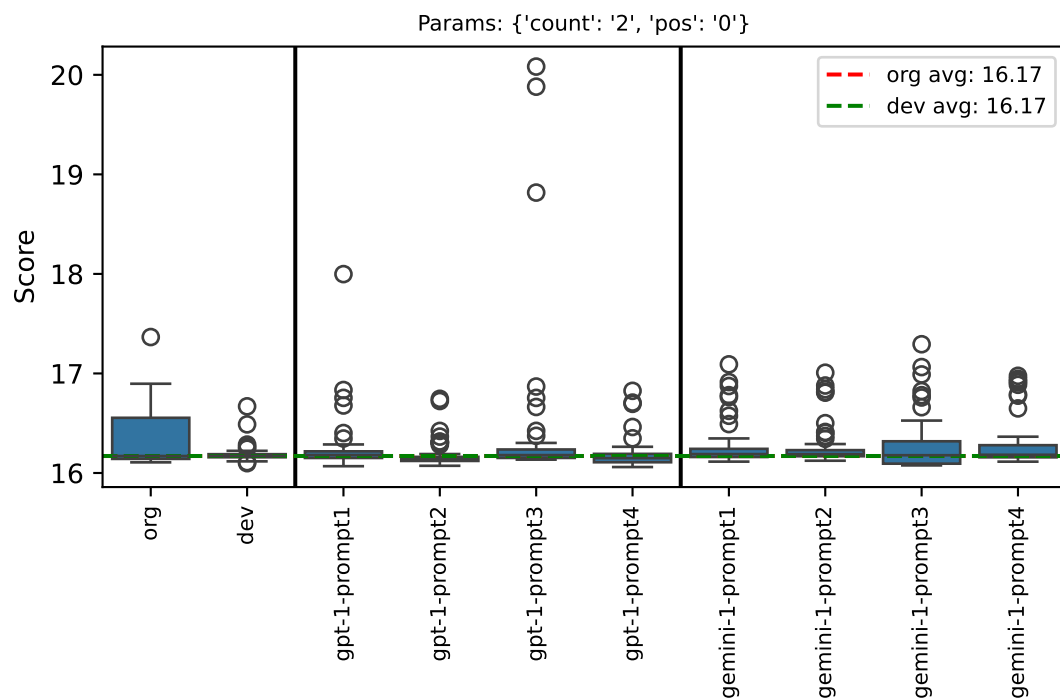
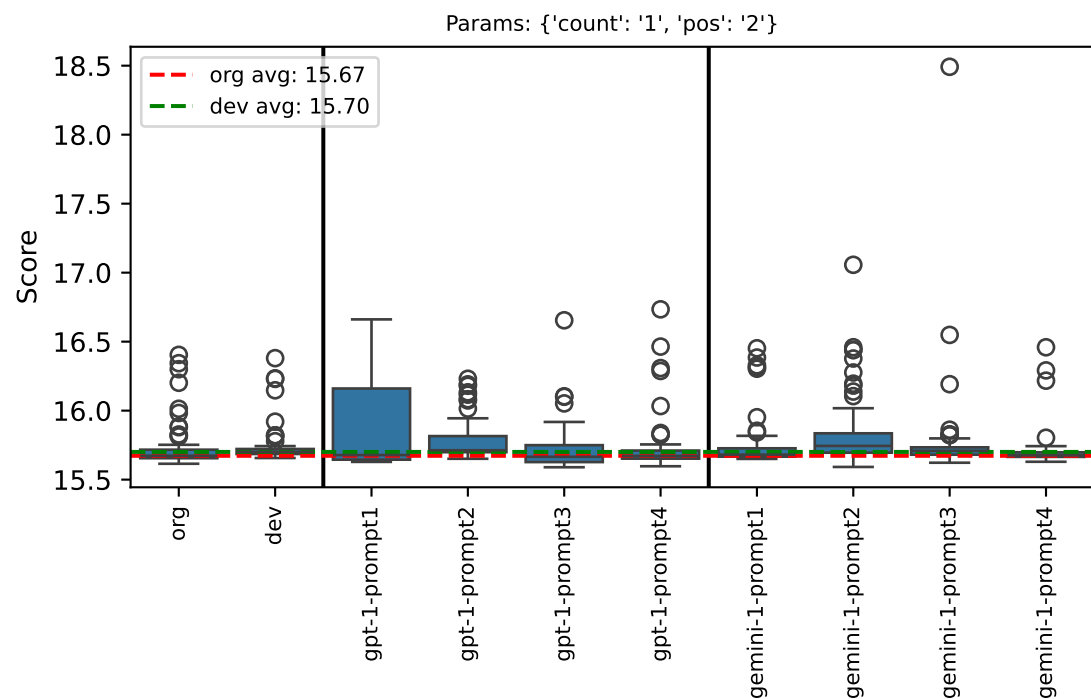
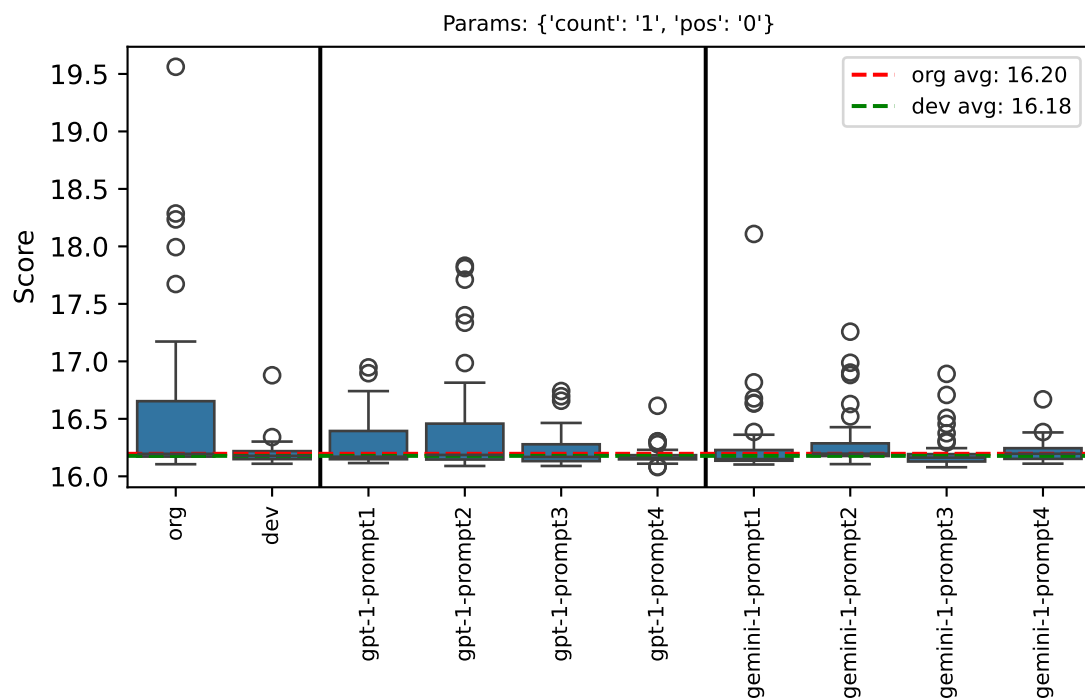
Params: {'count': '10000', 'n': '200'}

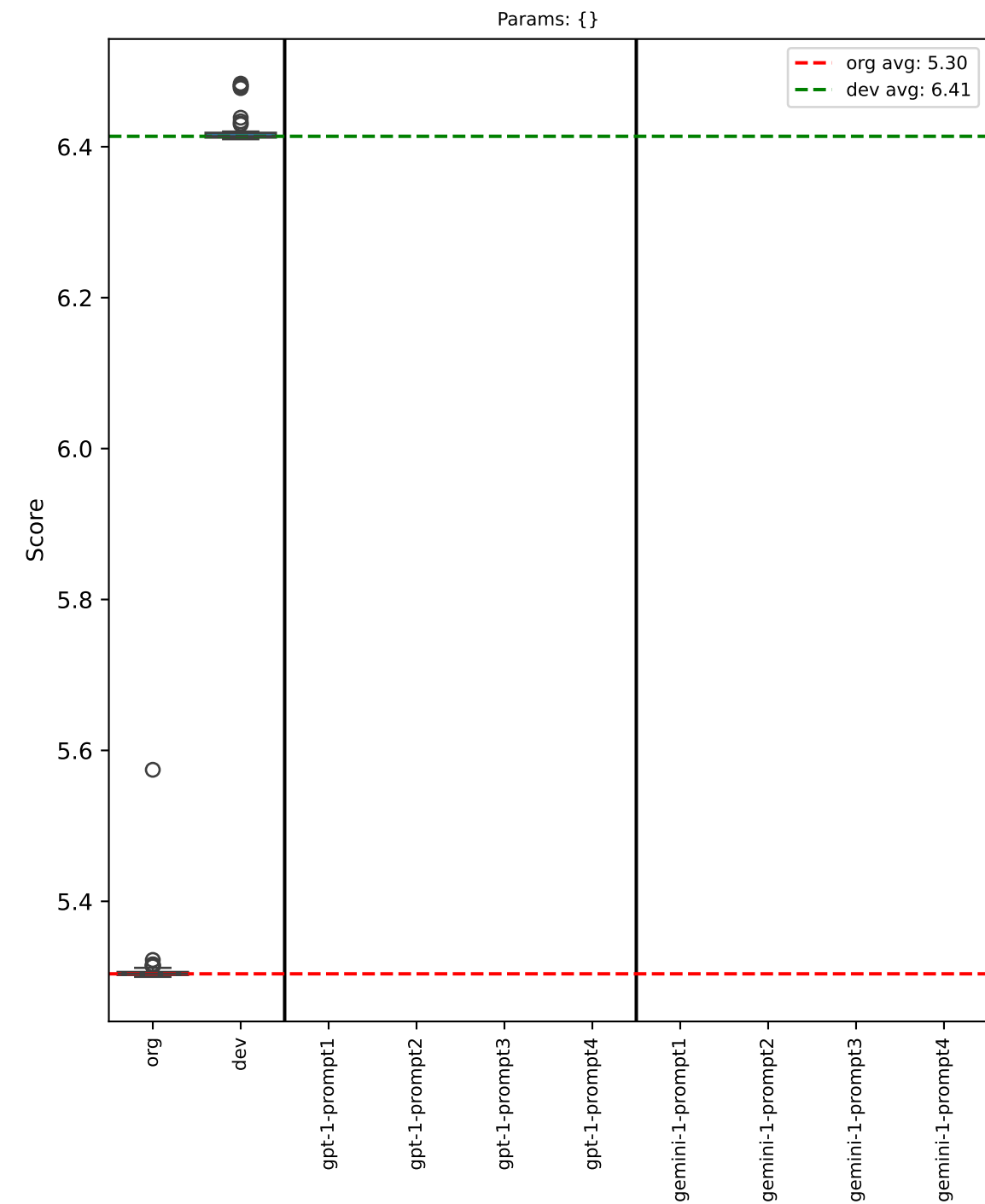


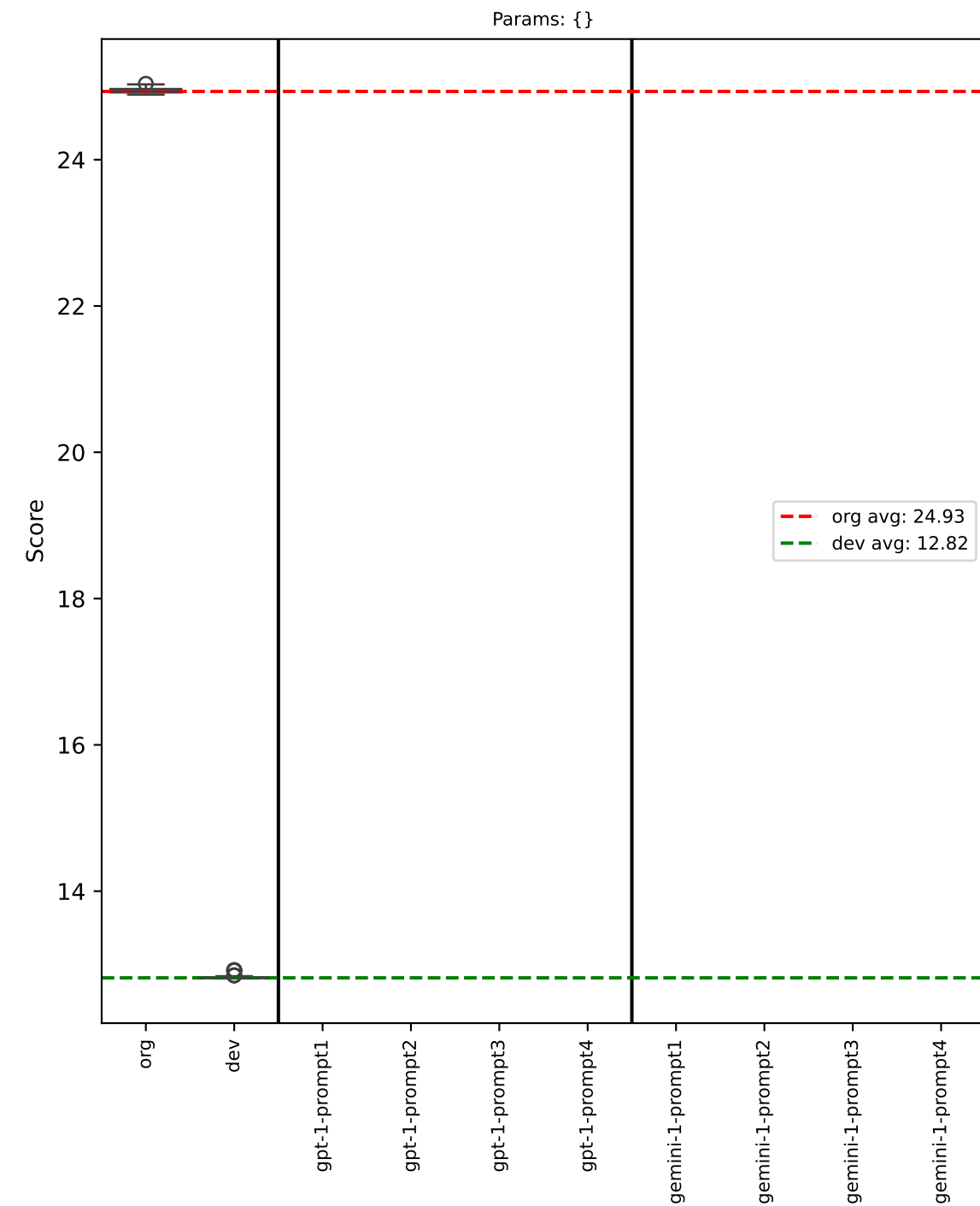
Benchmark: org.roaringbitmap.longlong.ShiftLeftFromSpecifiedPositionBenchmark.optimized

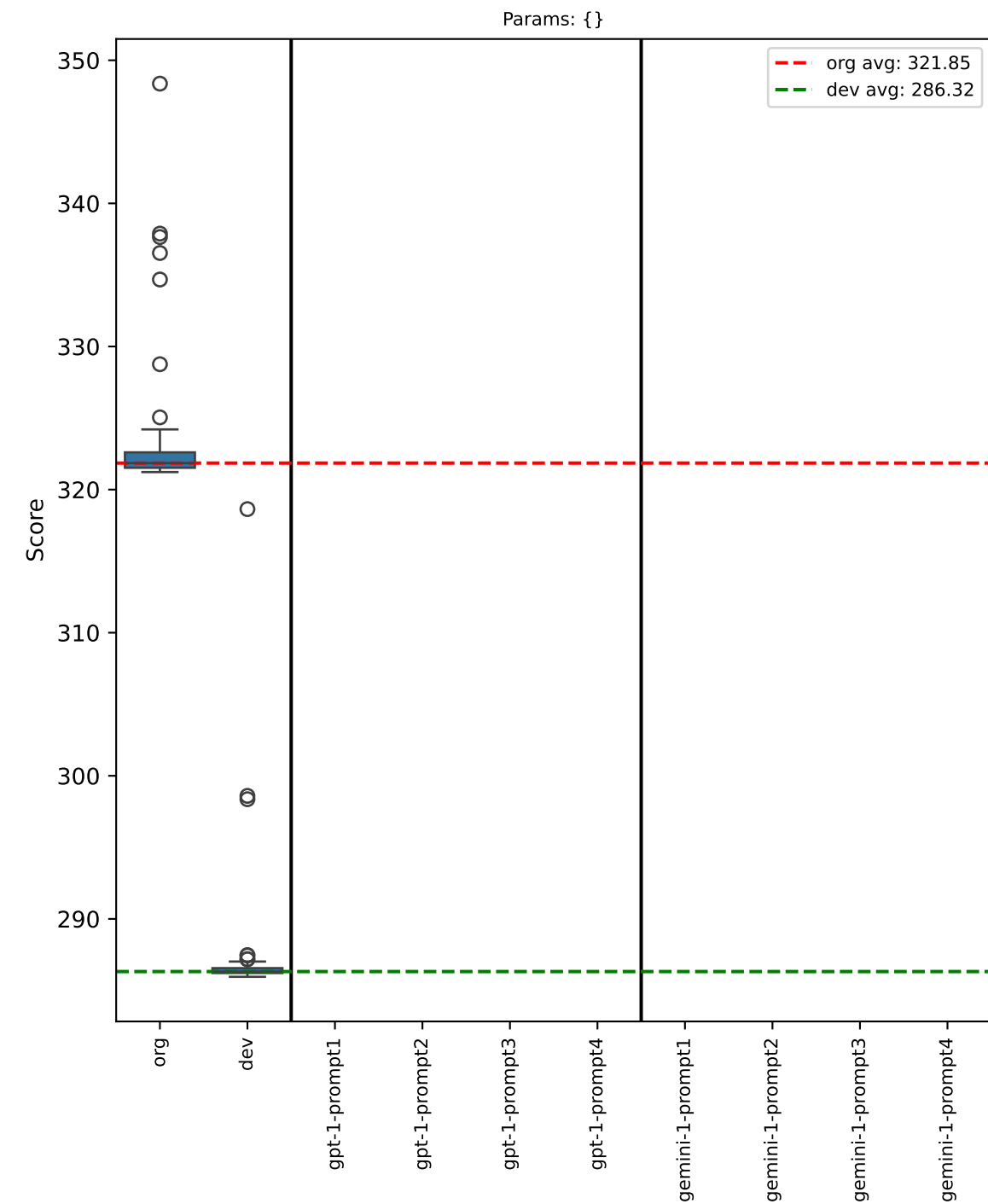


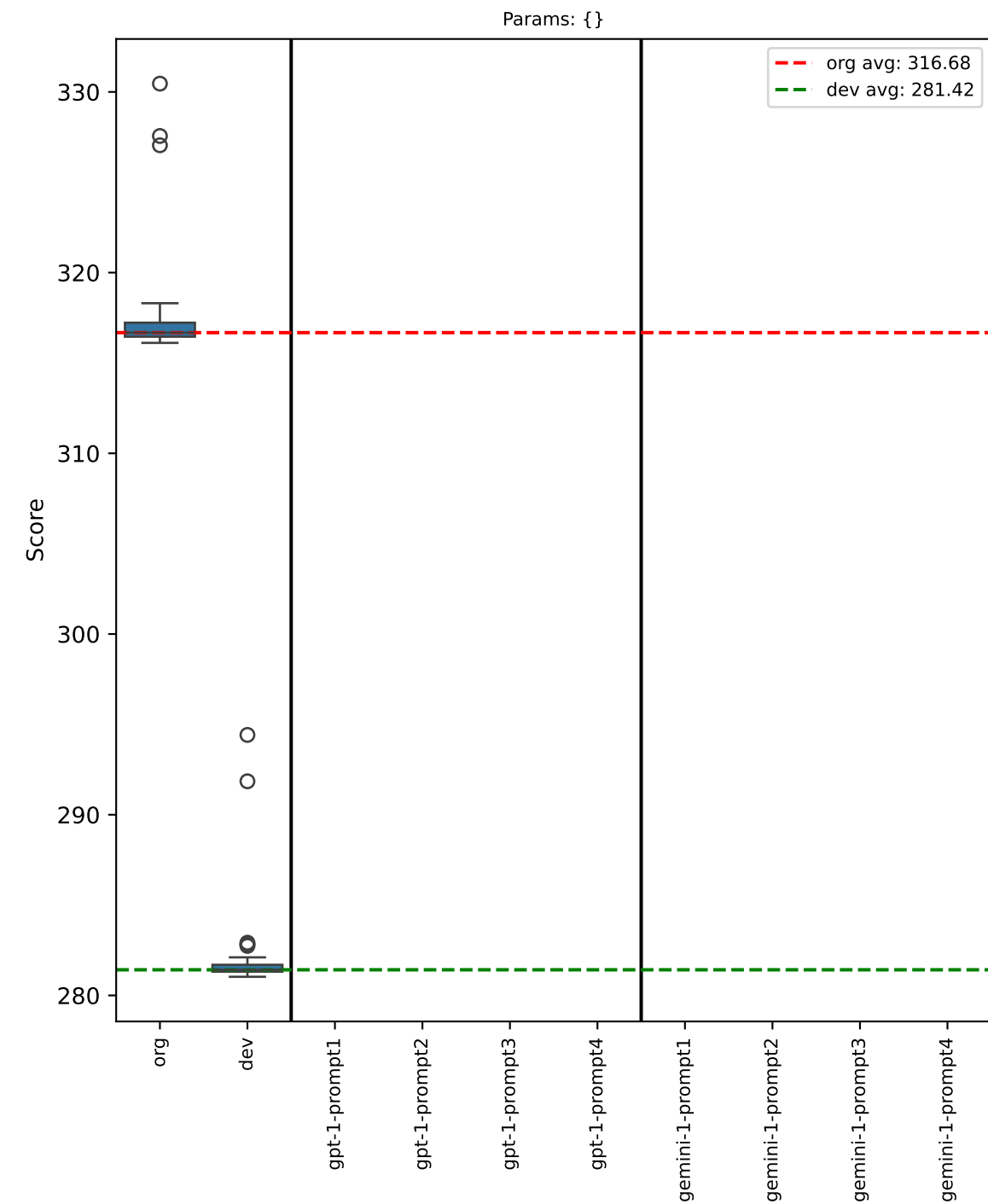
Benchmark: org.roaringbitmap.longlong.ShiftLeftFromSpecifiedPositionBenchmark.original





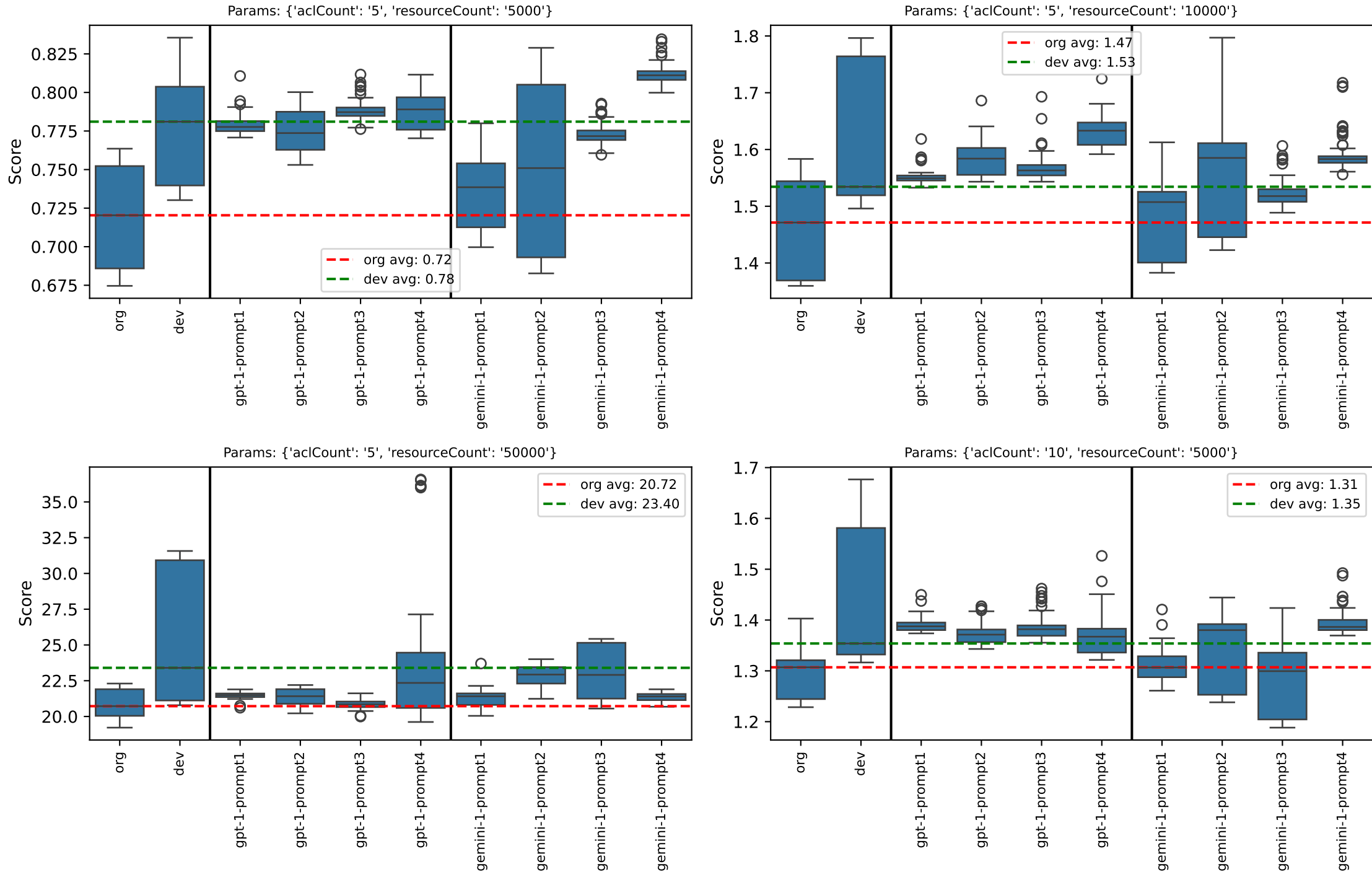






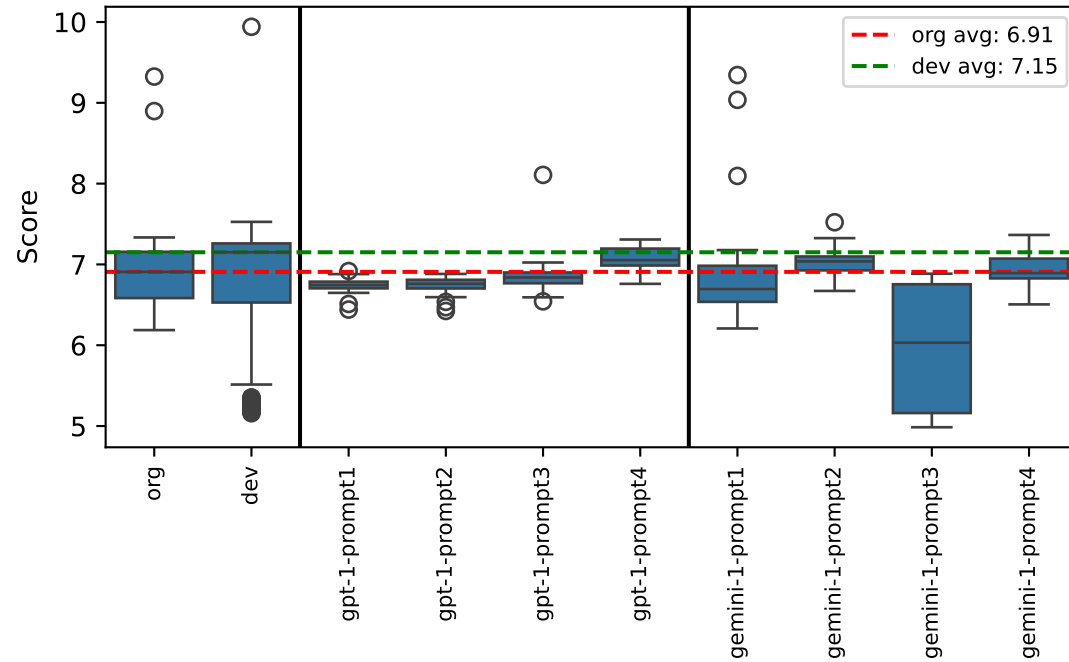
Project: kafka

Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAclsIterator

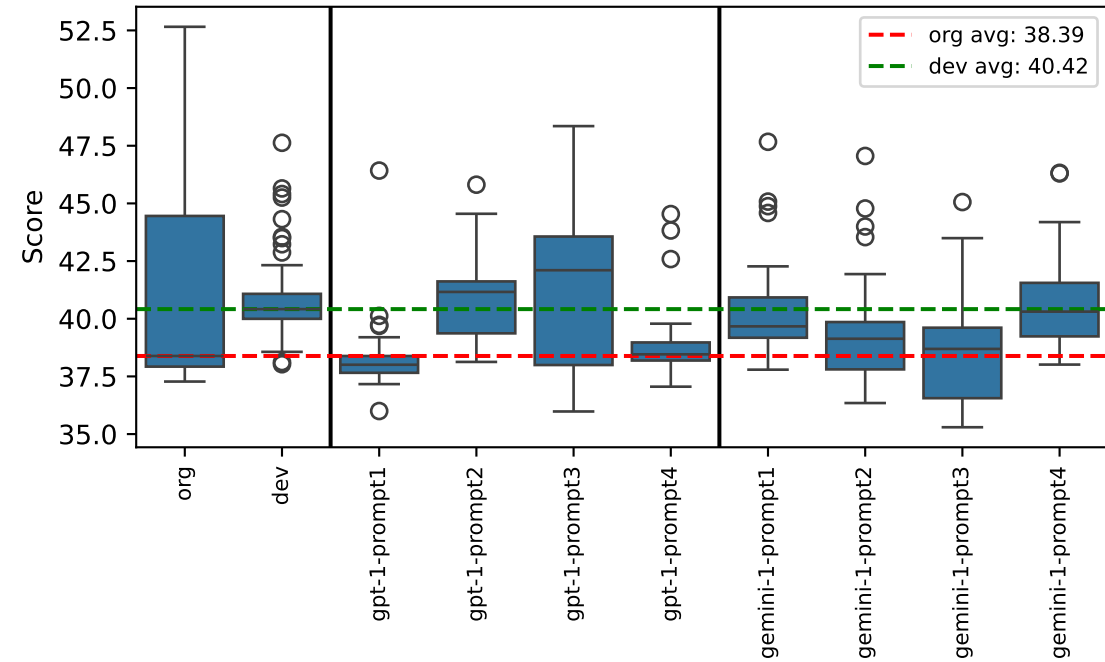


Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAclsIterator

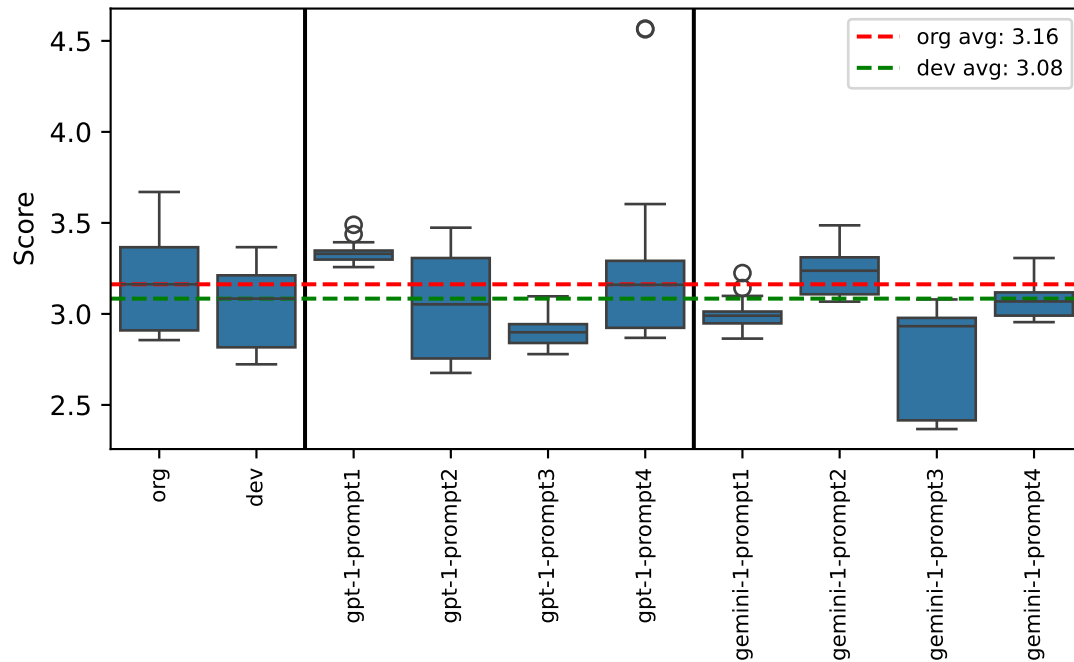
Params: {'aclCount': '10', 'resourceCount': '10000'}



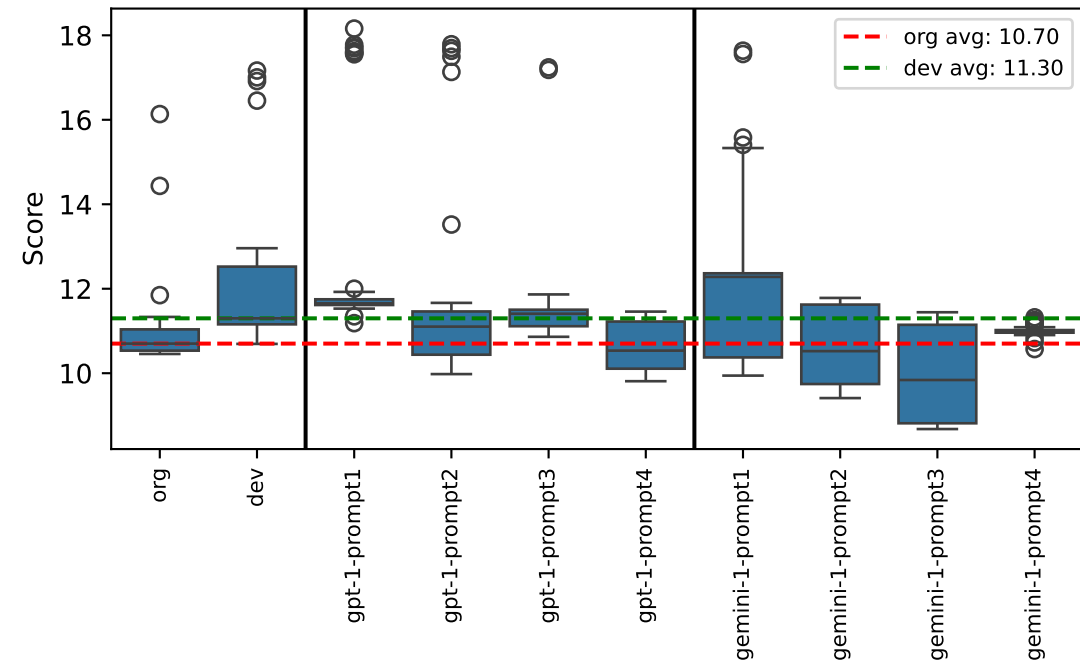
Params: {'aclCount': '10', 'resourceCount': '50000'}



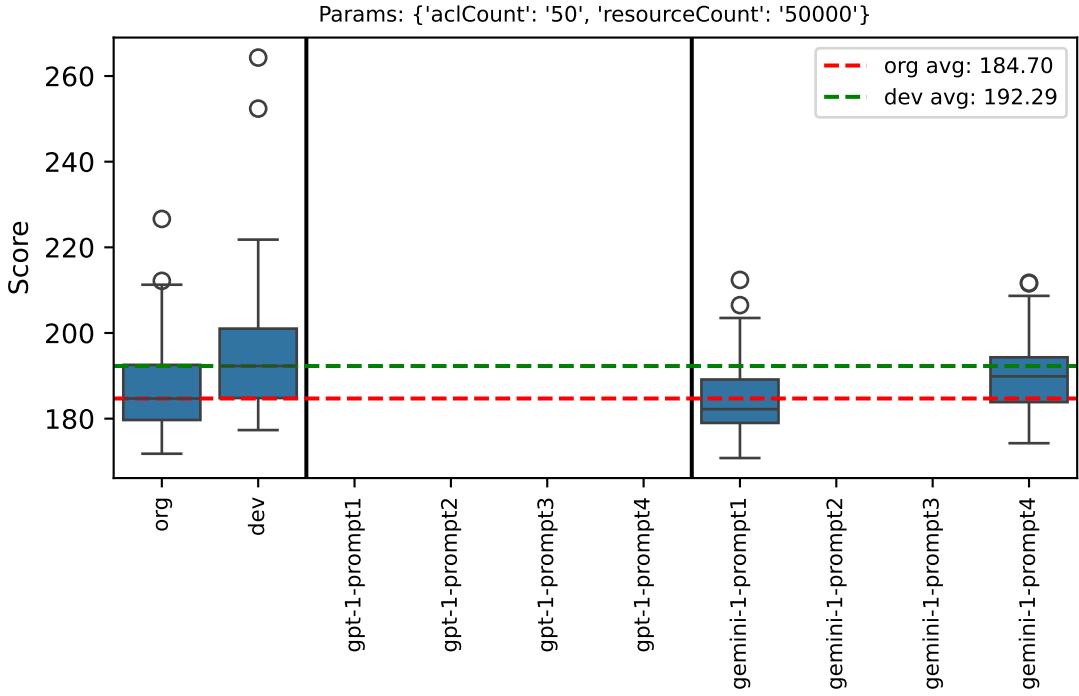
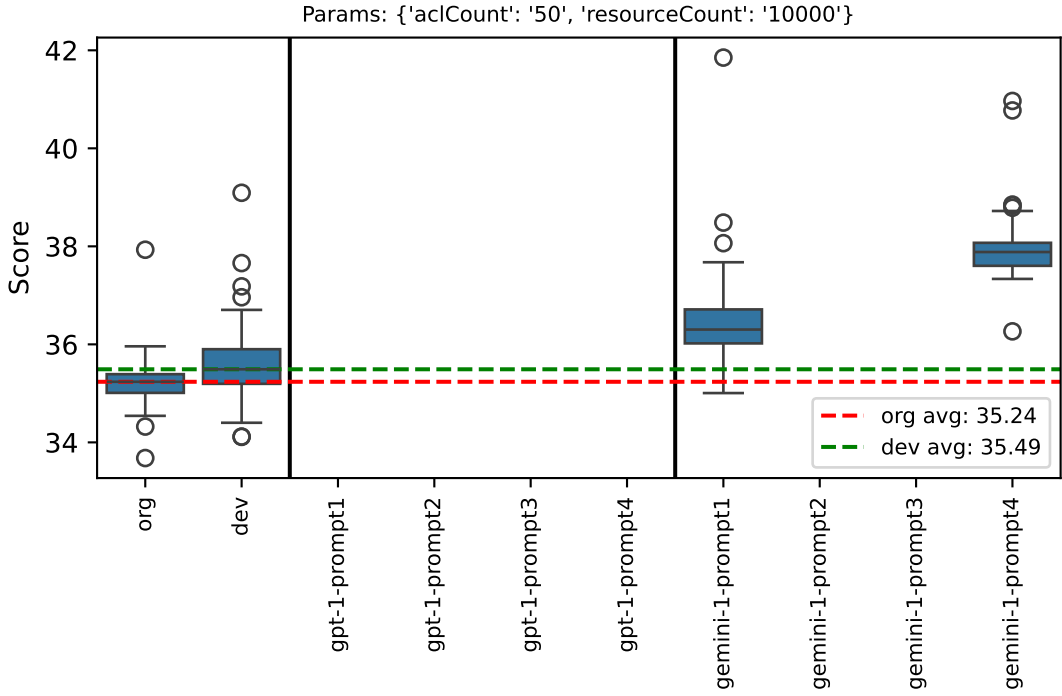
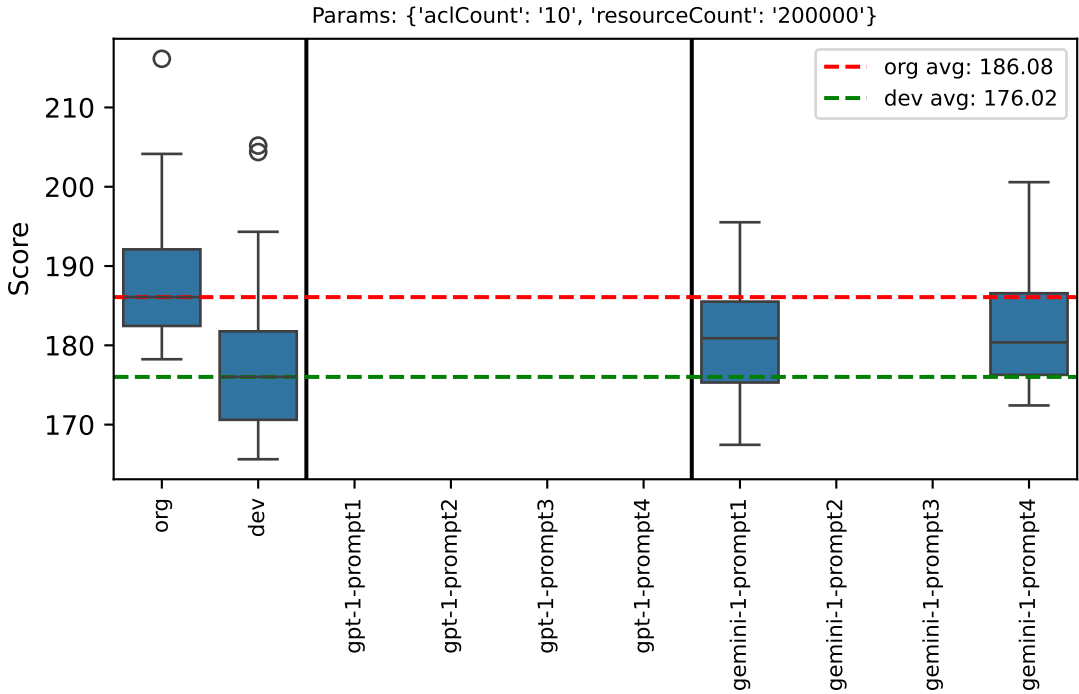
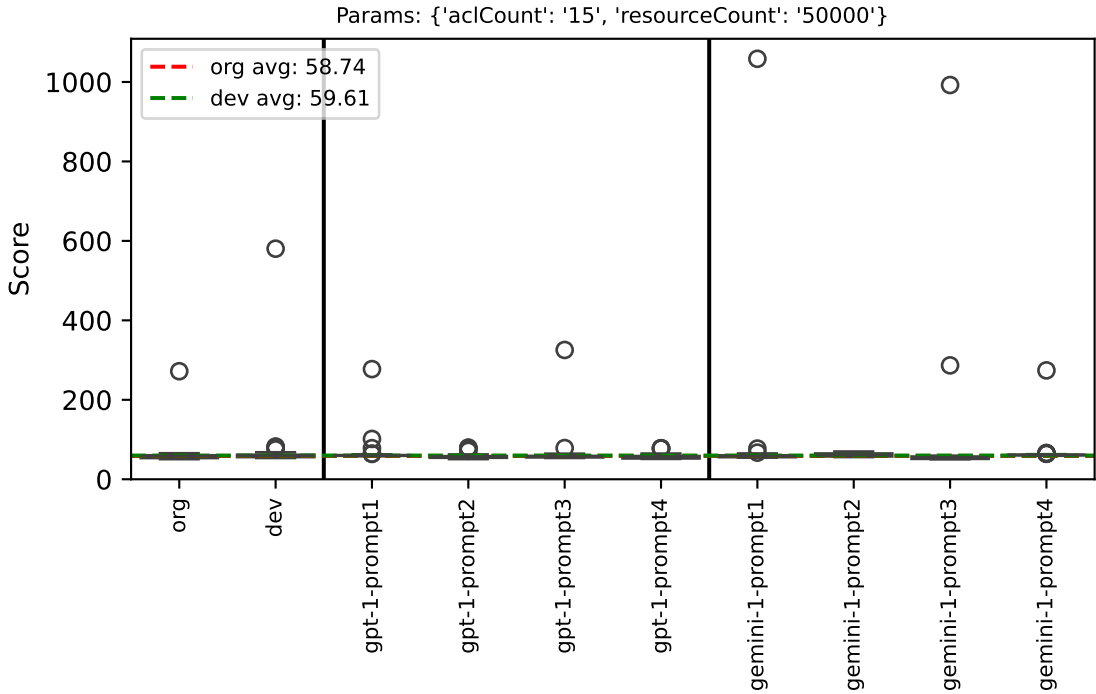
Params: {'aclCount': '15', 'resourceCount': '5000'}



Params: {'aclCount': '15', 'resourceCount': '10000'}

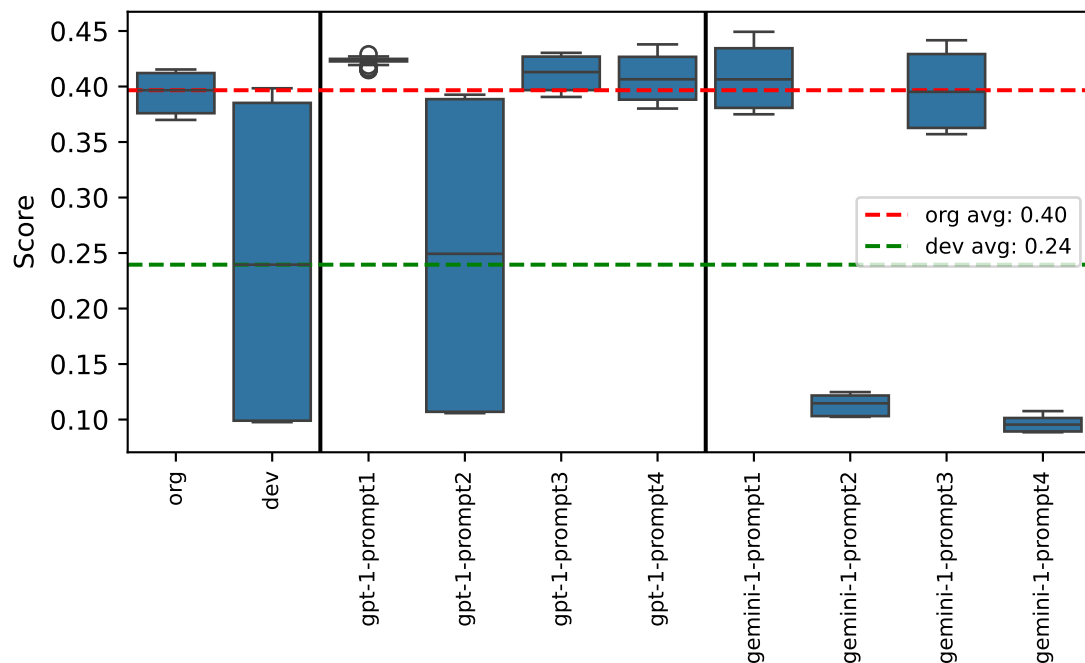


Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAclsIterator

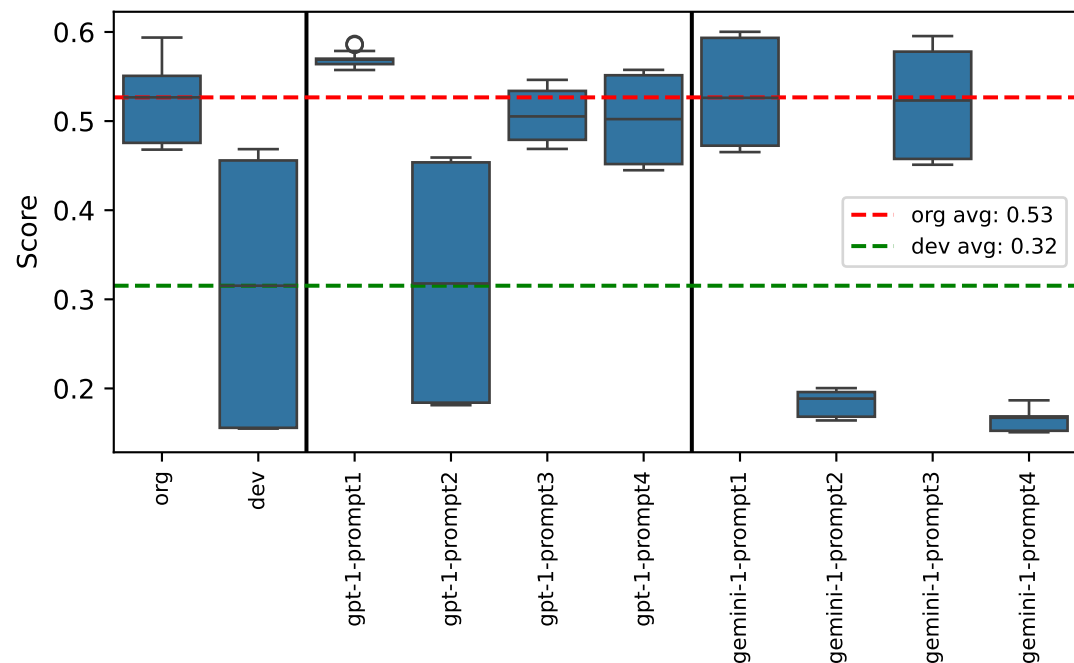


Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAuthorizer

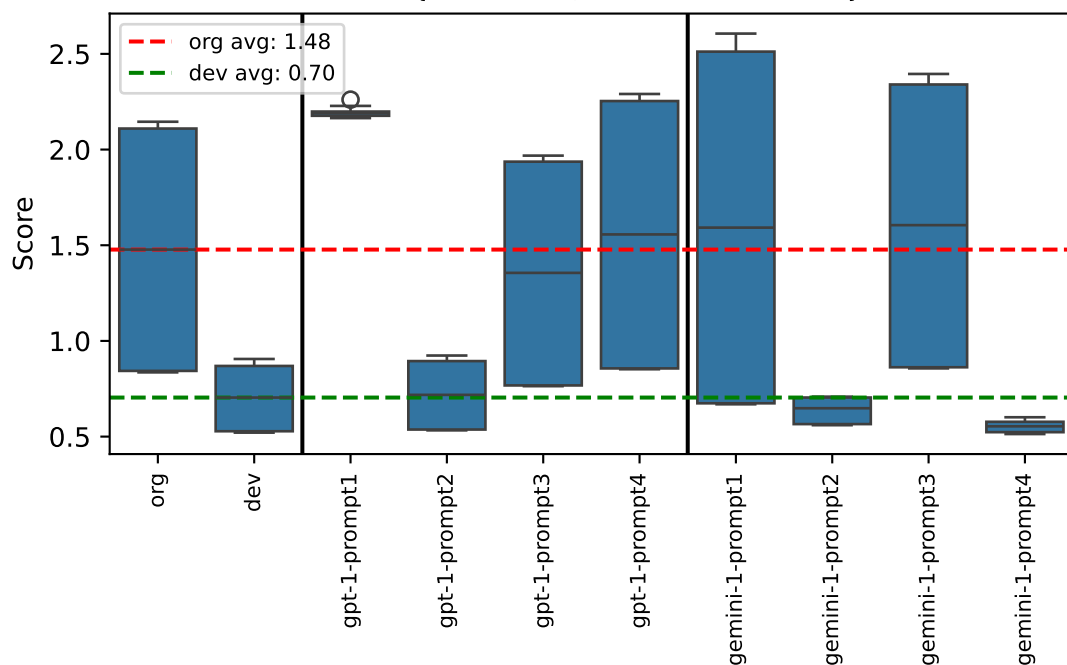
Params: {'aclCount': '5', 'resourceCount': '5000'}



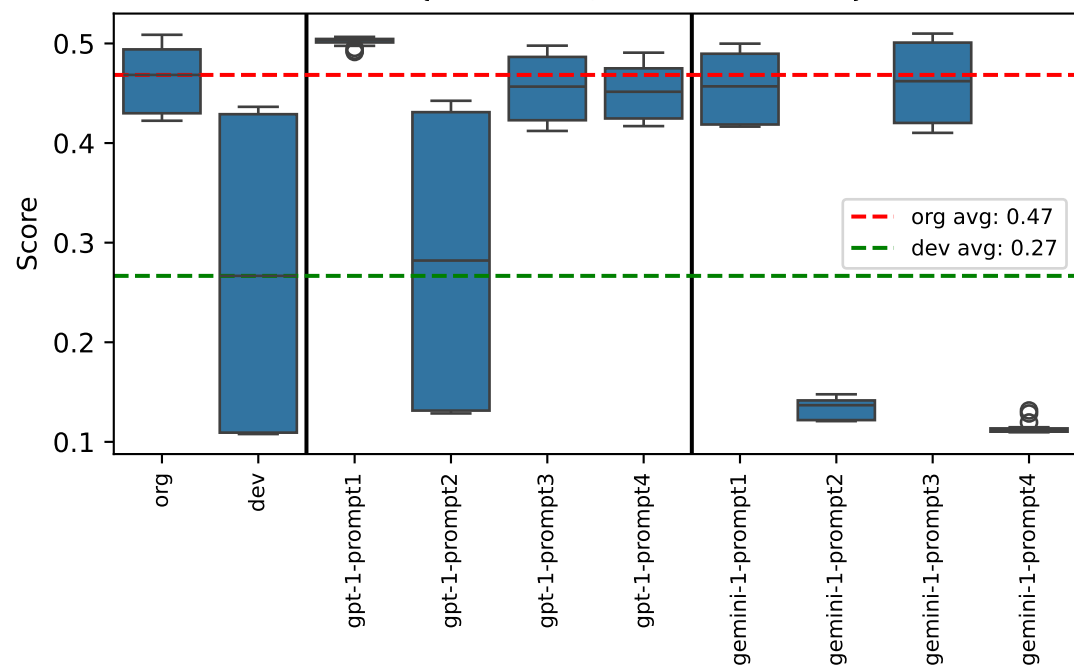
Params: {'aclCount': '5', 'resourceCount': '10000'}



Params: {'aclCount': '5', 'resourceCount': '50000'}

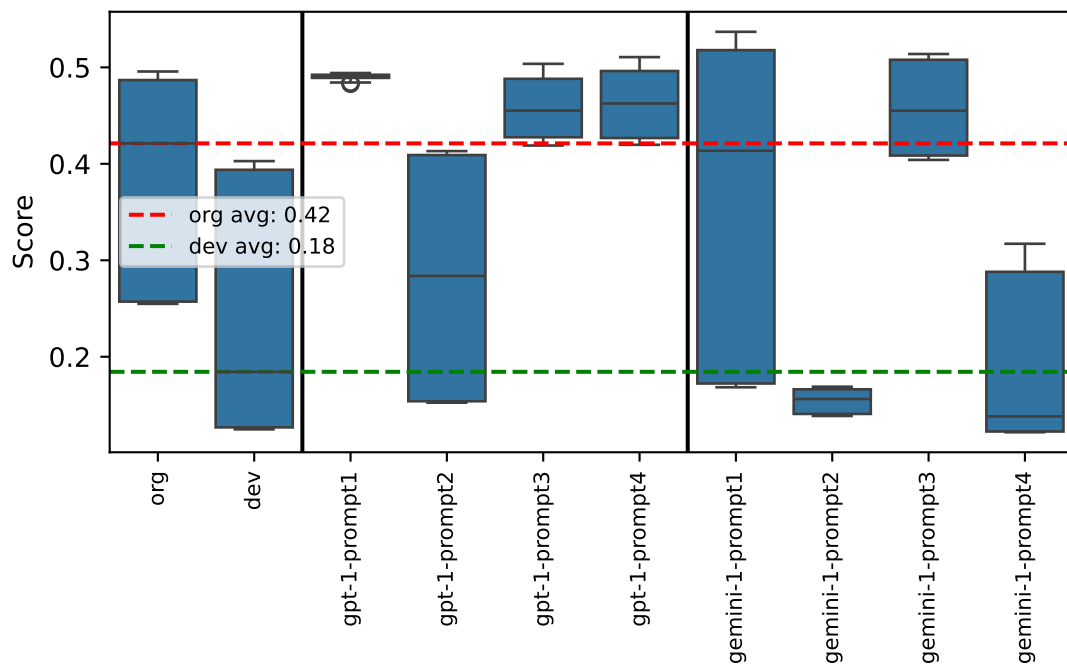


Params: {'aclCount': '10', 'resourceCount': '5000'}

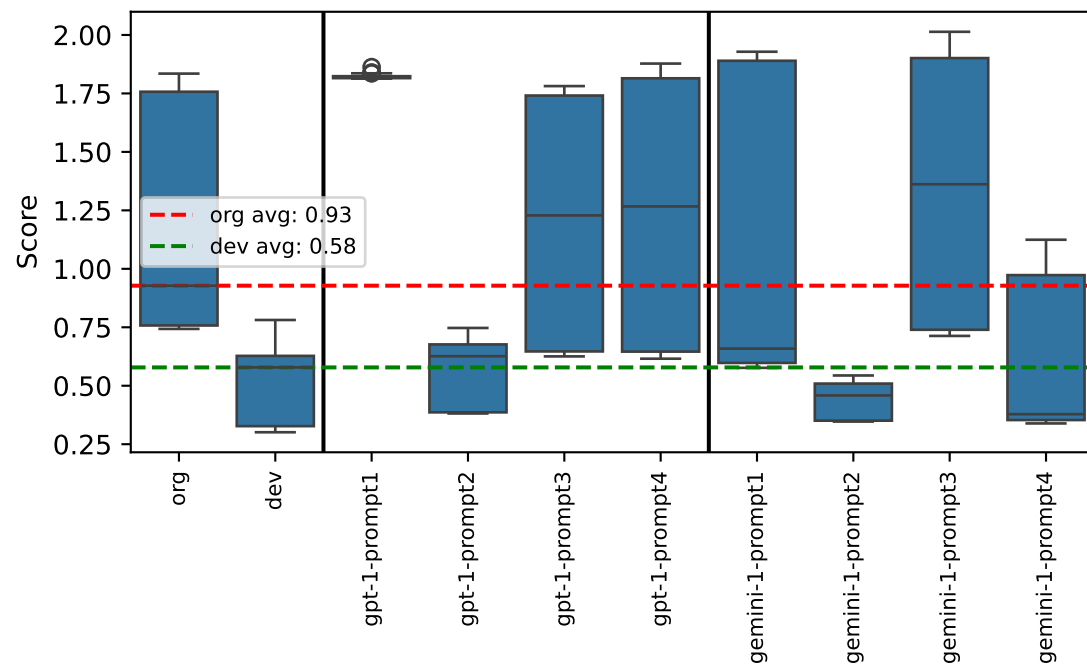


Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAuthorizer

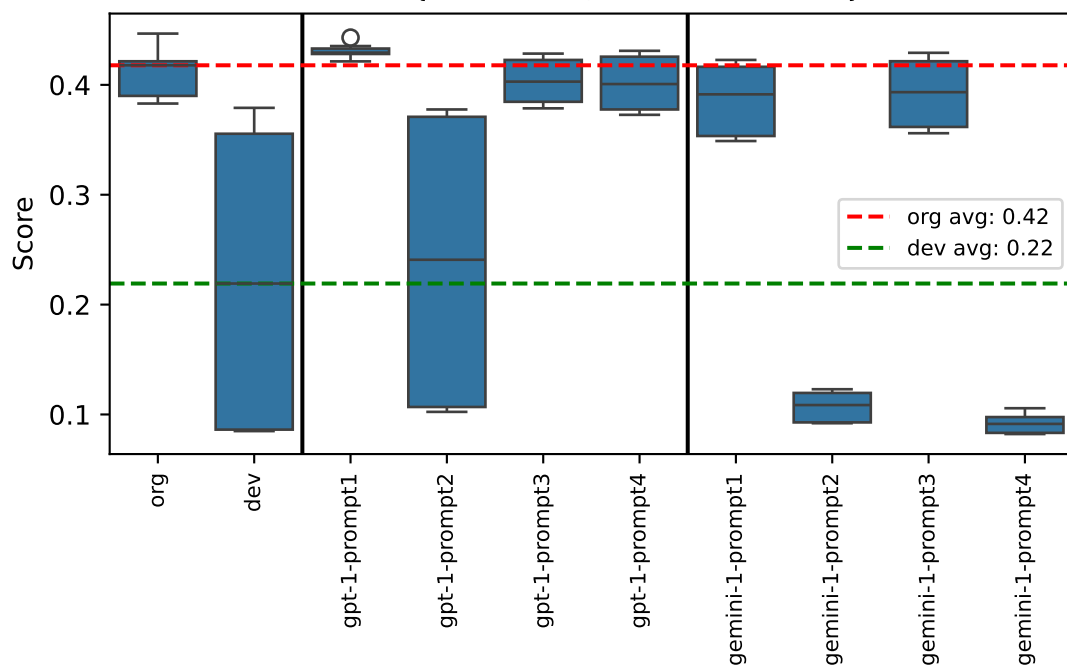
Params: {'aclCount': '10', 'resourceCount': '10000'}



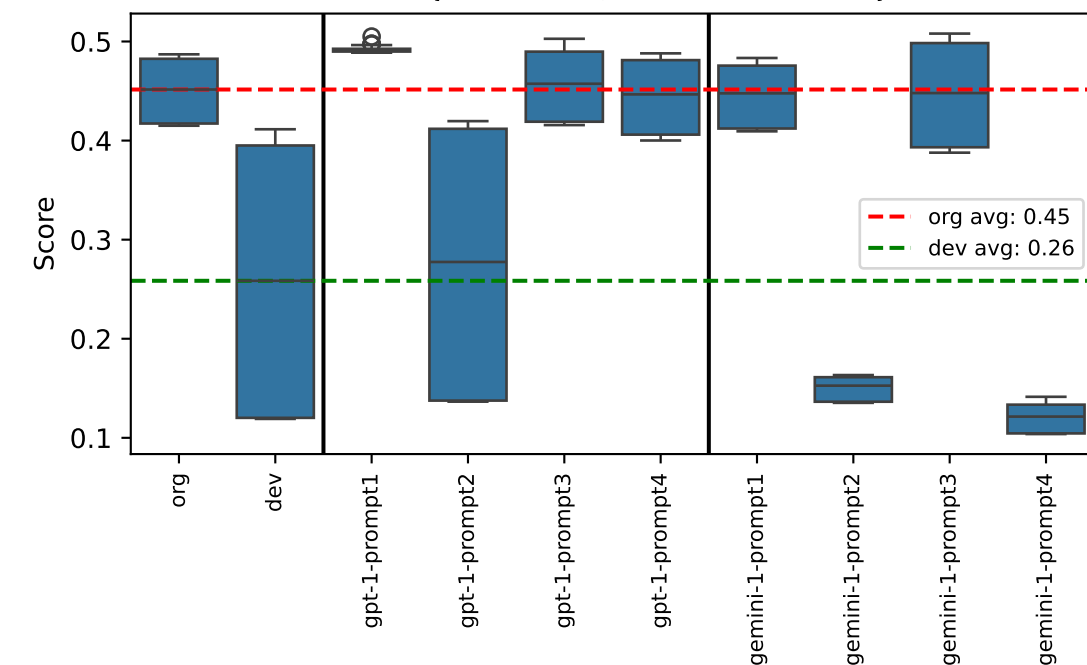
Params: {'aclCount': '10', 'resourceCount': '50000'}



Params: {'aclCount': '15', 'resourceCount': '5000'}

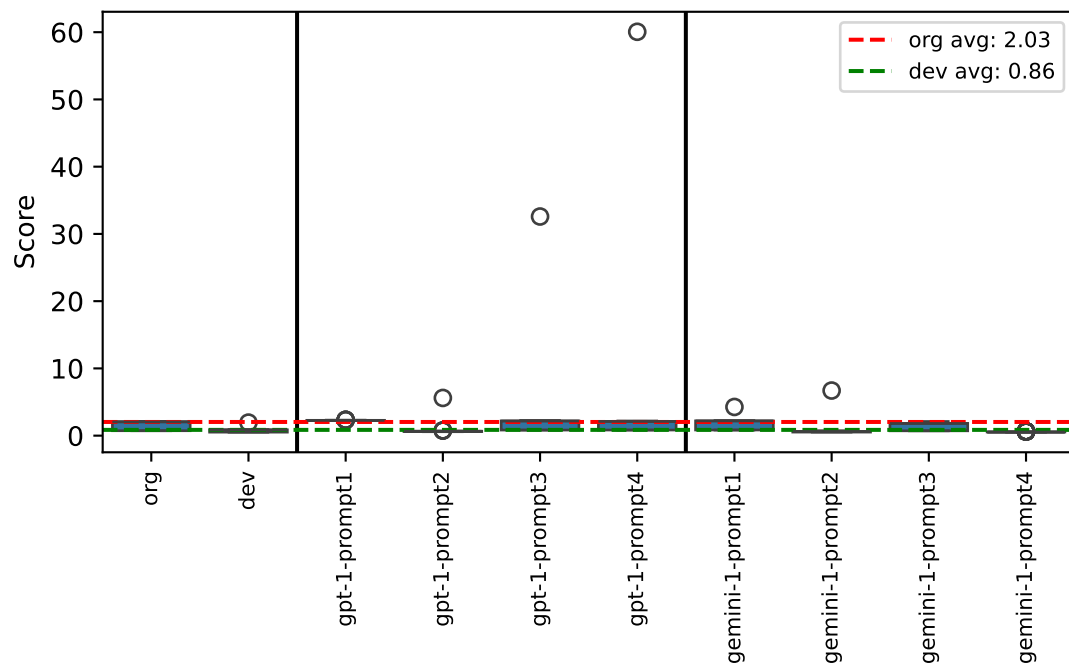


Params: {'aclCount': '15', 'resourceCount': '10000'}

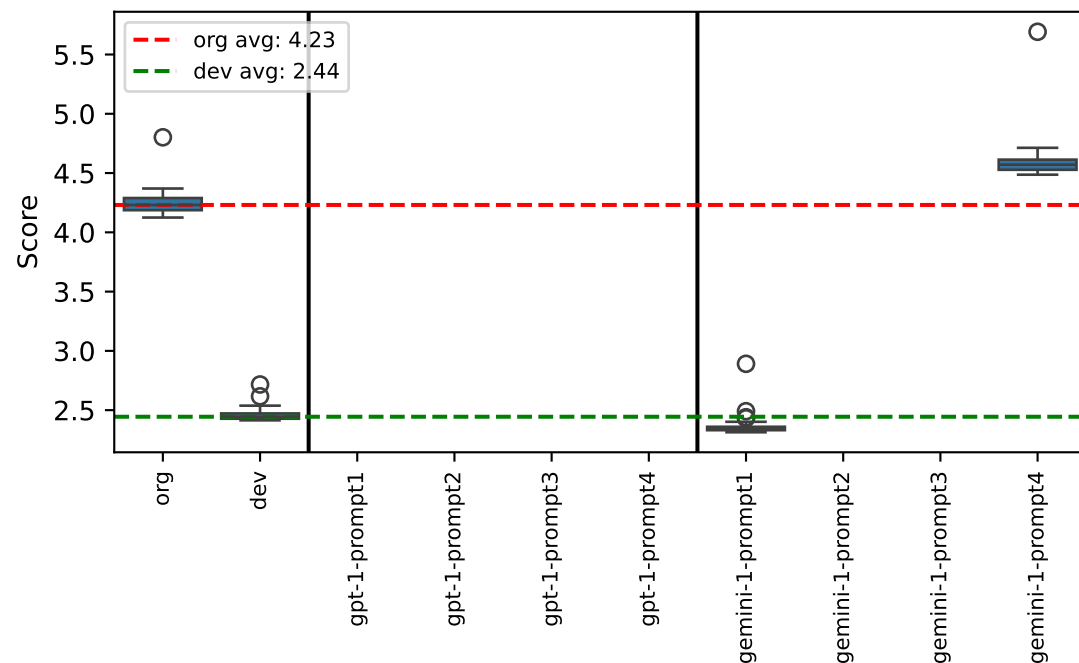


Benchmark: org.apache.kafka.jmh.acl.AclAuthorizerBenchmark.testAuthorizer

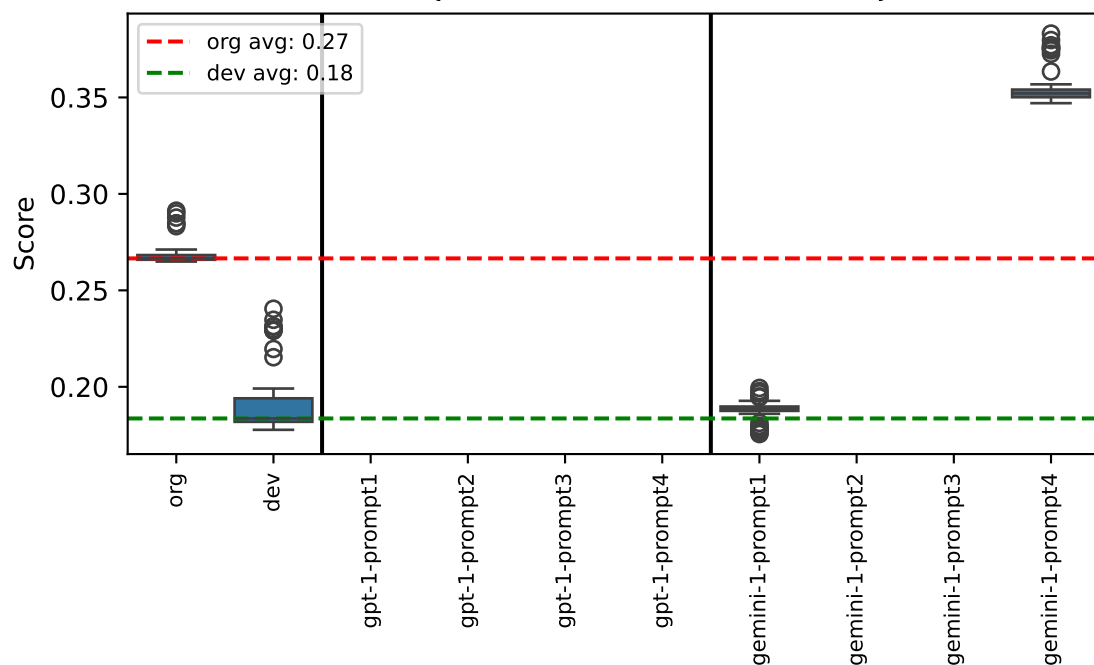
Params: {'aclCount': '15', 'resourceCount': '50000'}



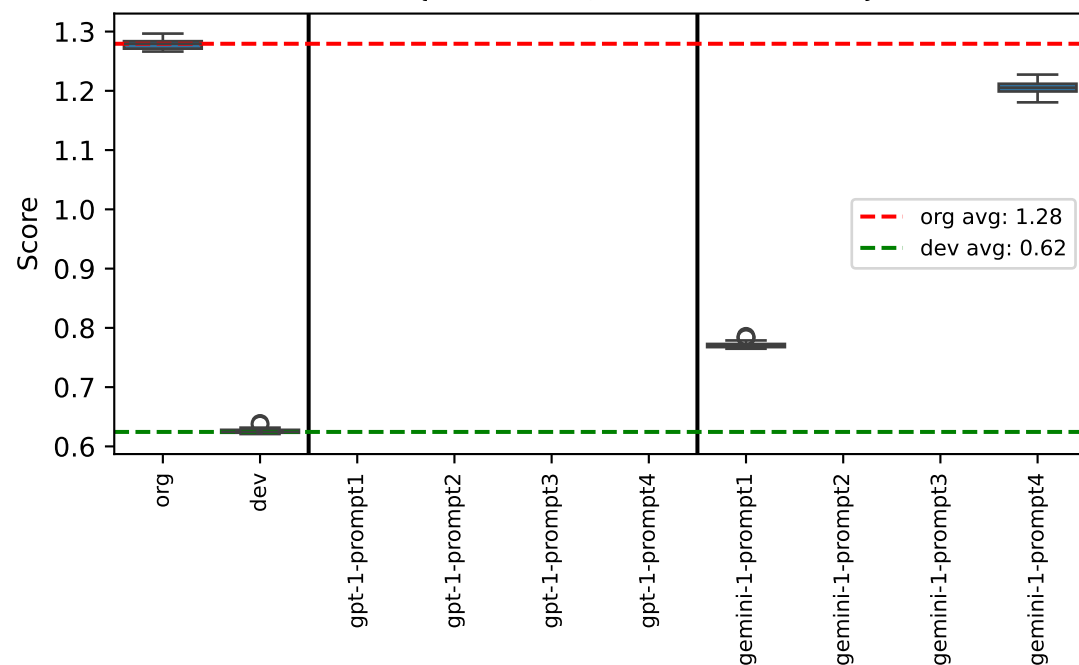
Params: {'aclCount': '10', 'resourceCount': '200000'}



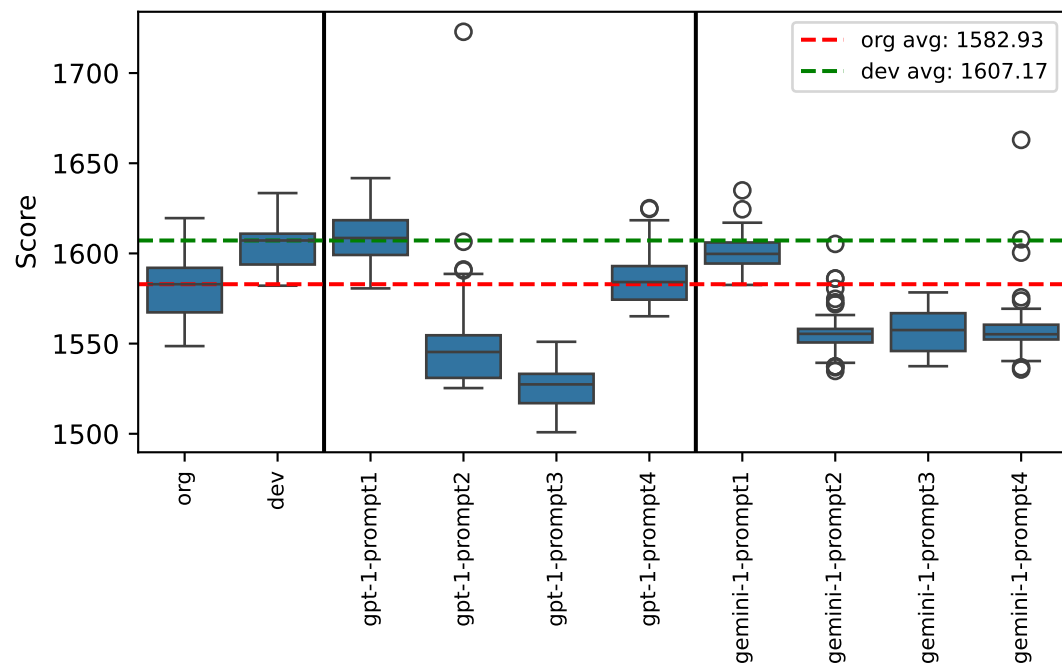
Params: {'aclCount': '50', 'resourceCount': '10000'}



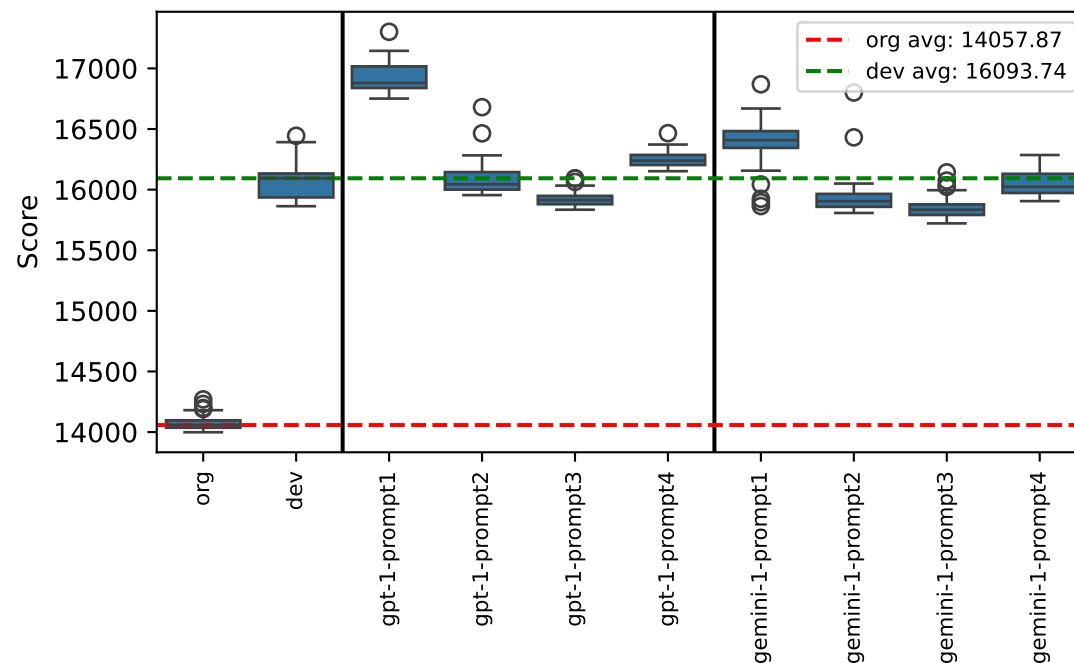
Params: {'aclCount': '50', 'resourceCount': '50000'}



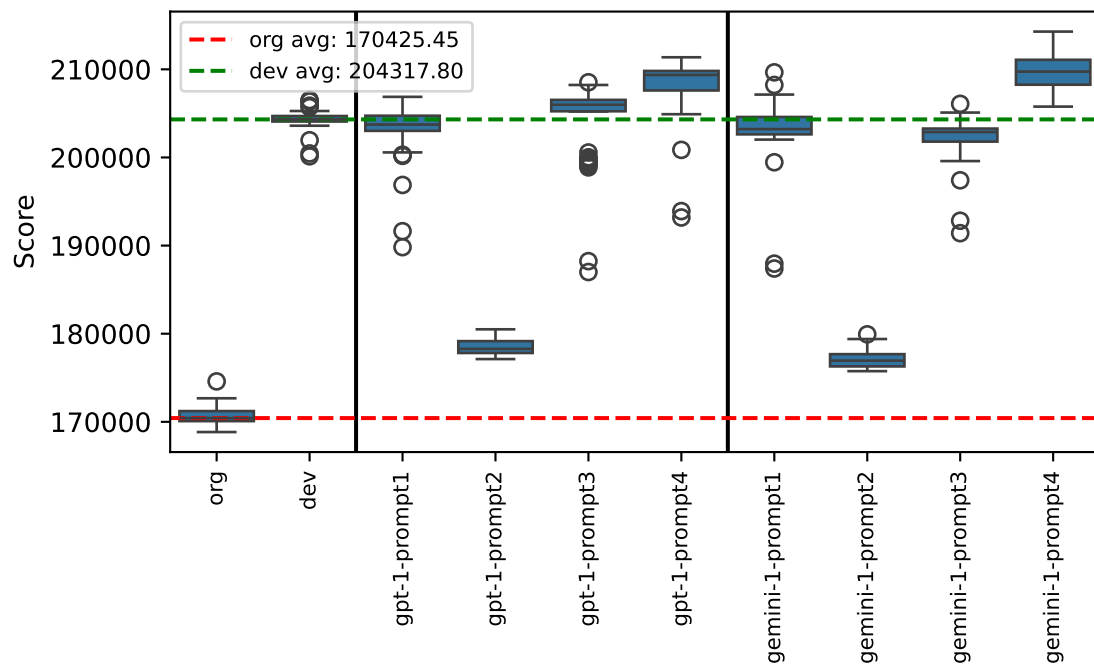
Params: {'includeExcludeFieldCount': '1', 'valueFieldCount': '100'}



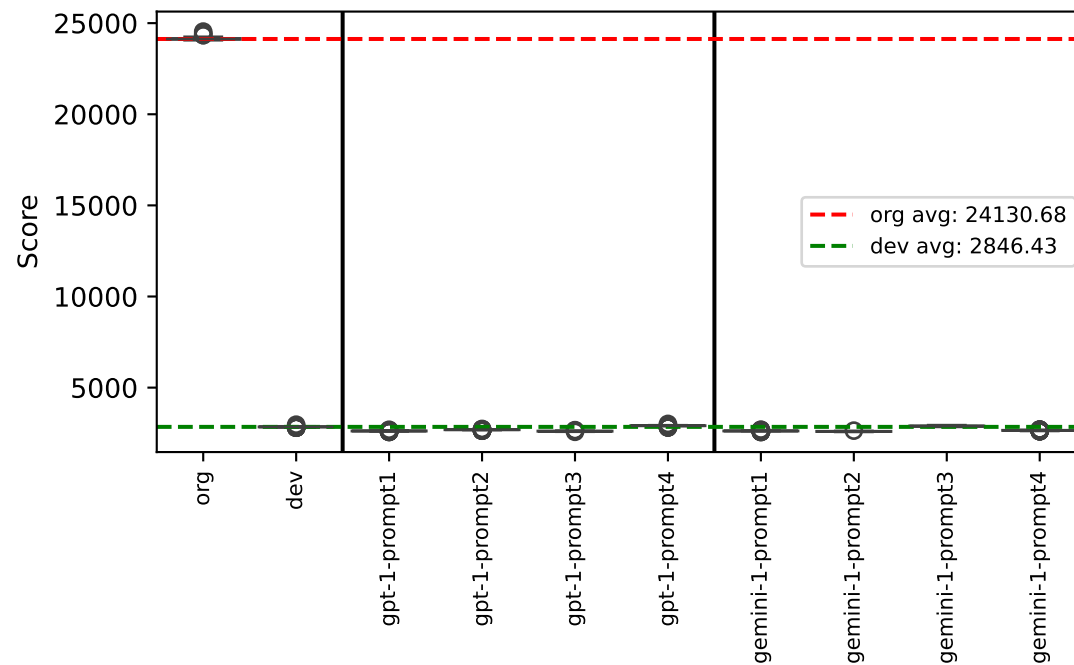
Params: {'includeExcludeFieldCount': '1', 'valueFieldCount': '1000'}

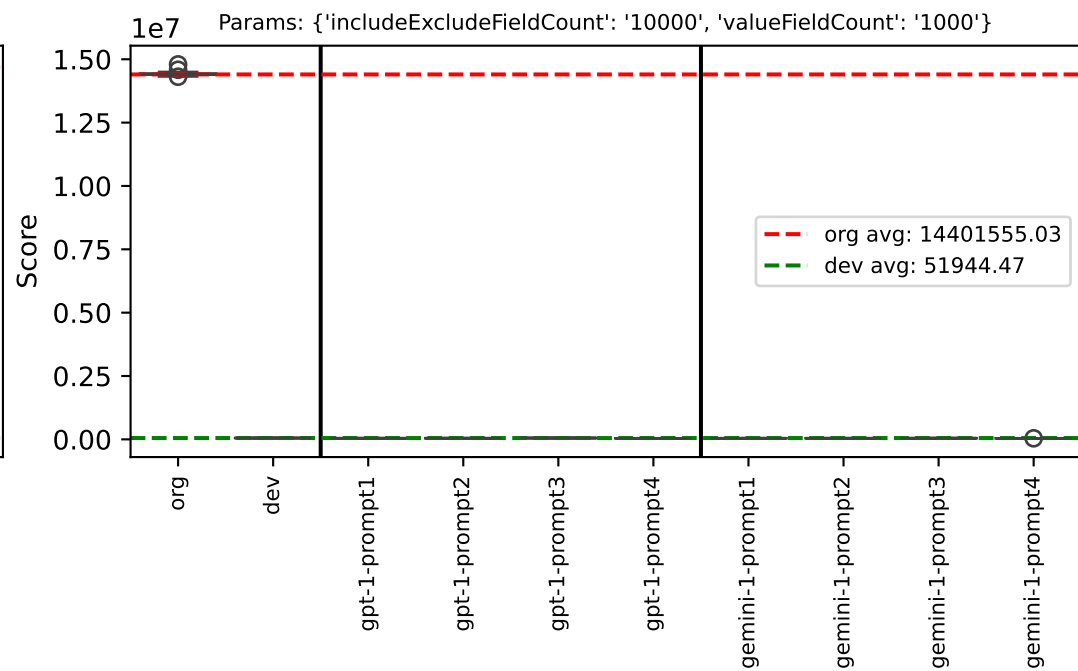
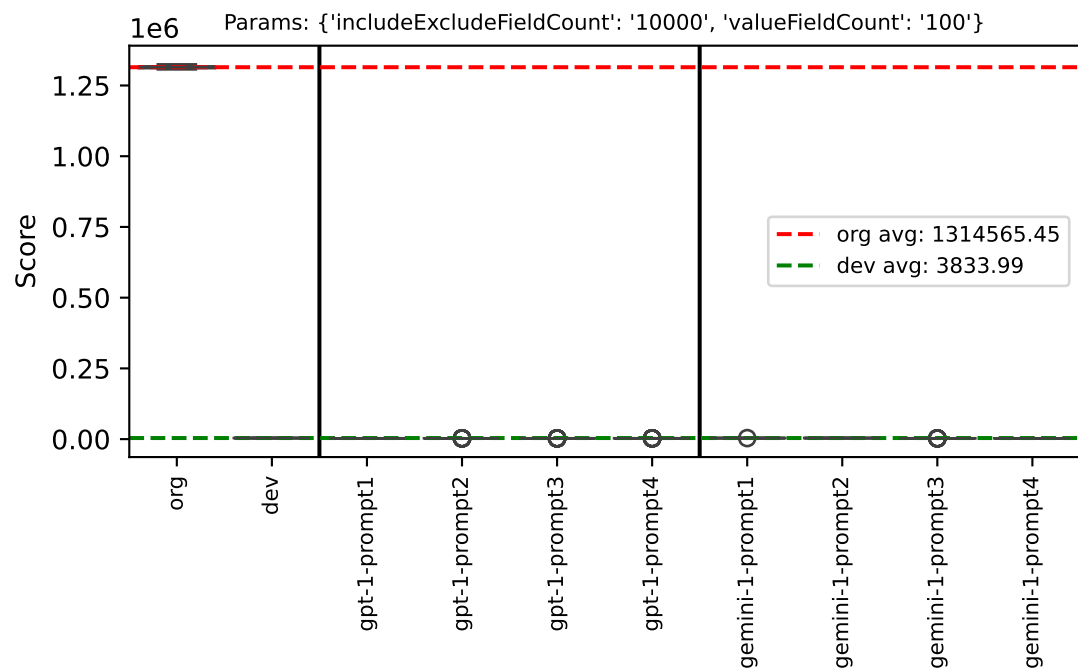
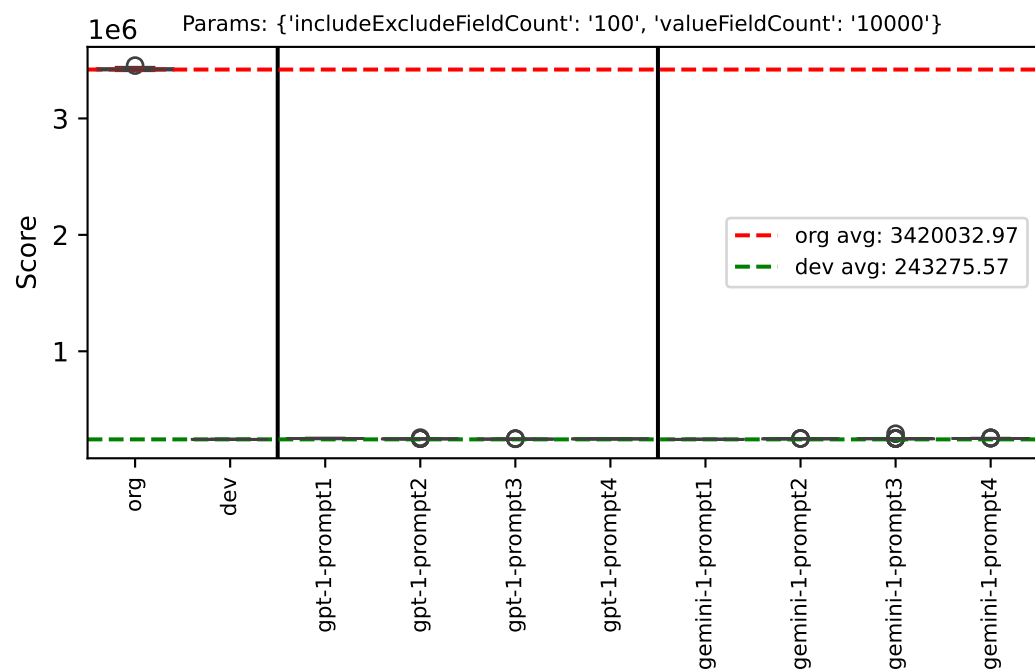
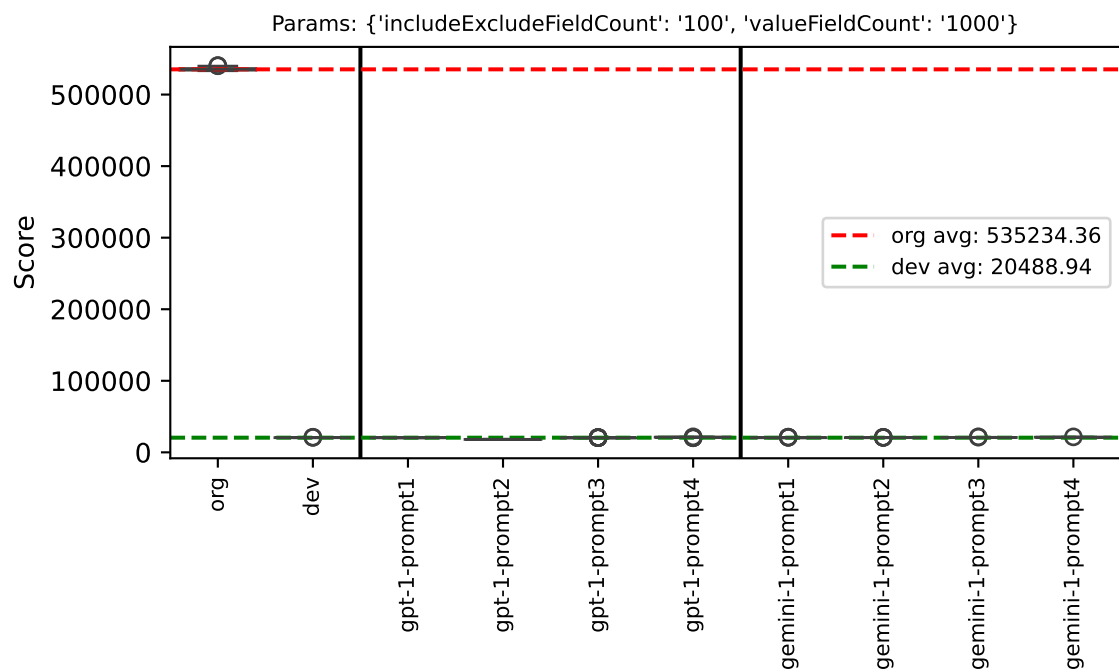


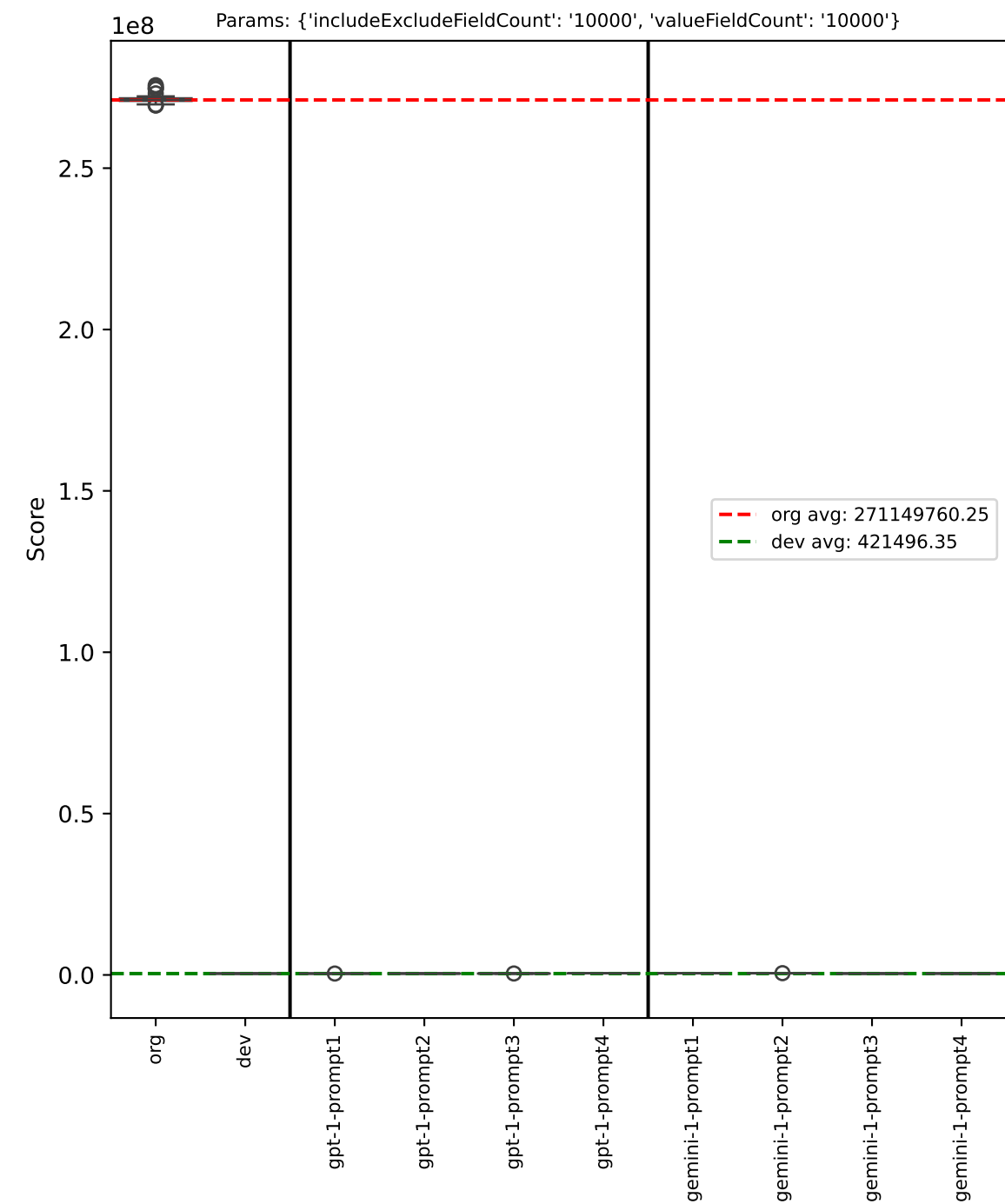
Params: {'includeExcludeFieldCount': '1', 'valueFieldCount': '10000'}



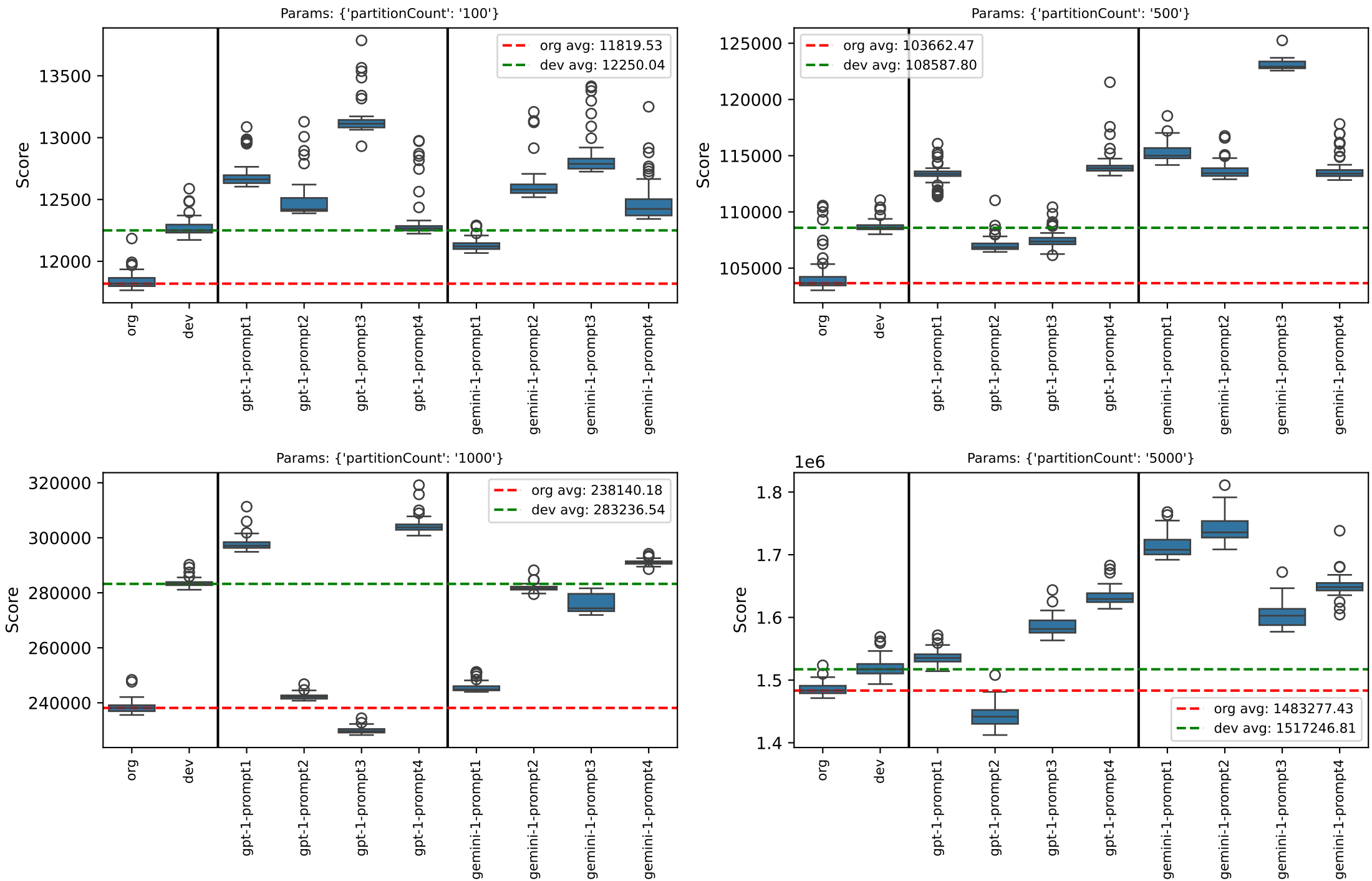
Params: {'includeExcludeFieldCount': '100', 'valueFieldCount': '100'}



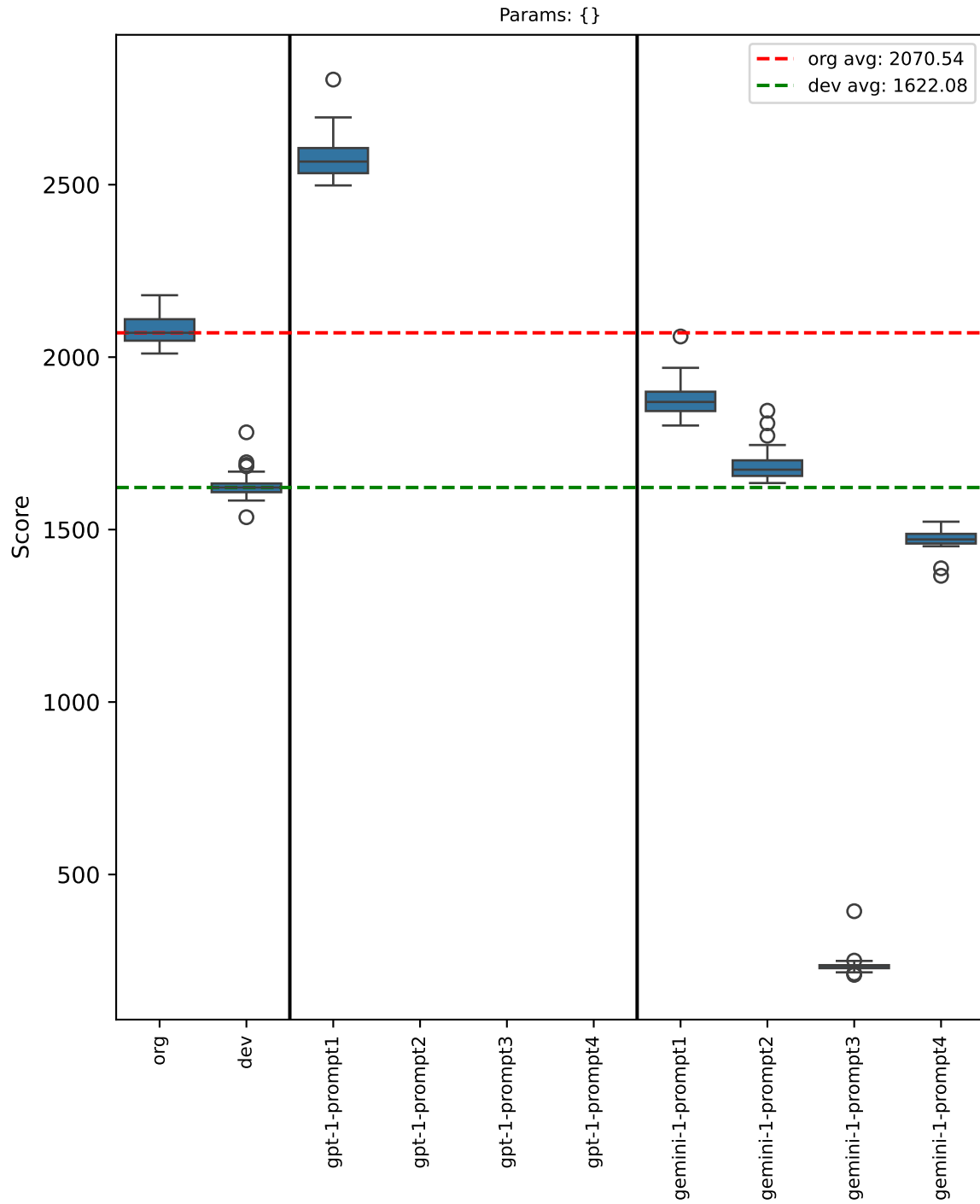




Benchmark: org.apache.kafka.jmh.fetcher.ReplicaFetcherThreadBenchmark.testFetcher

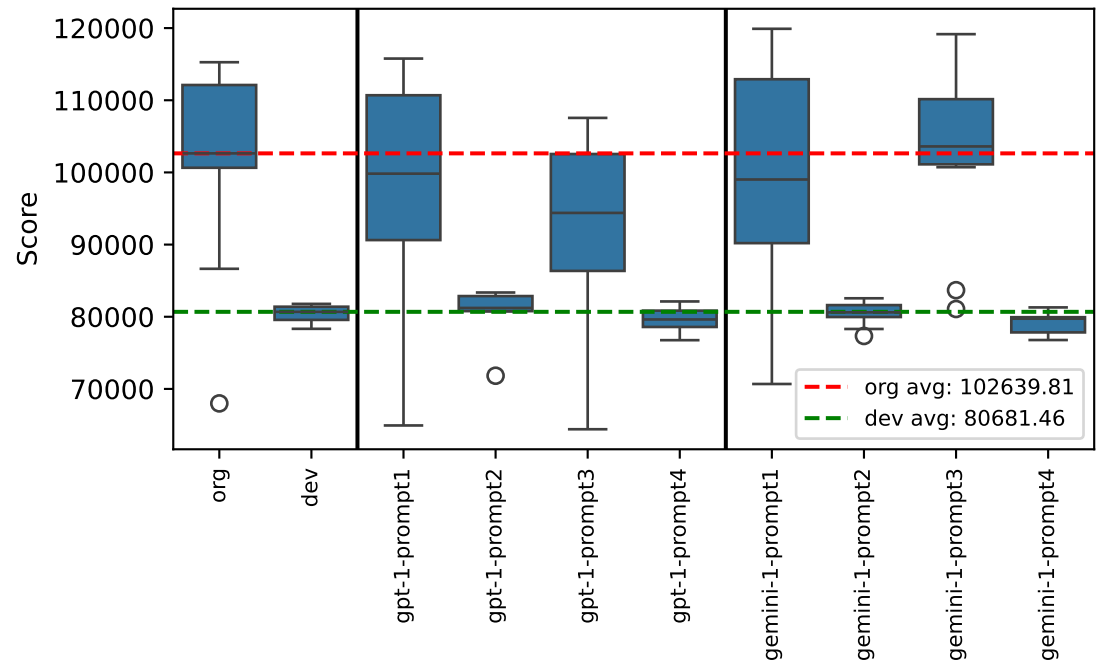
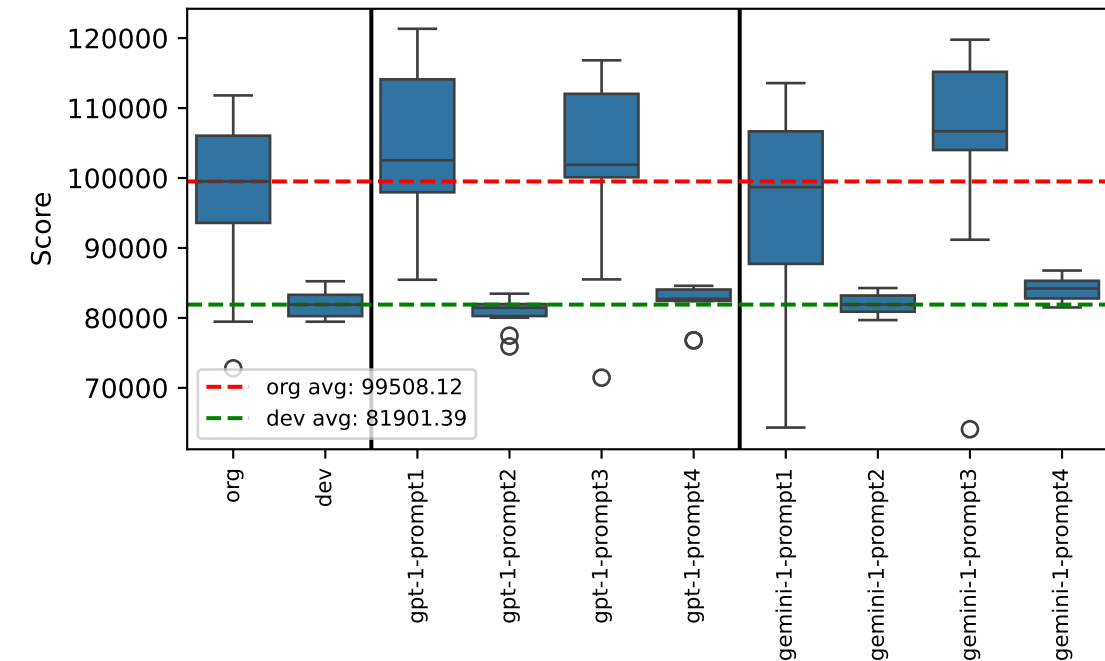


Benchmark: org.apache.kafka.jmh.partition.PartitionMakeFollowerBenchmark.testMakeFollower

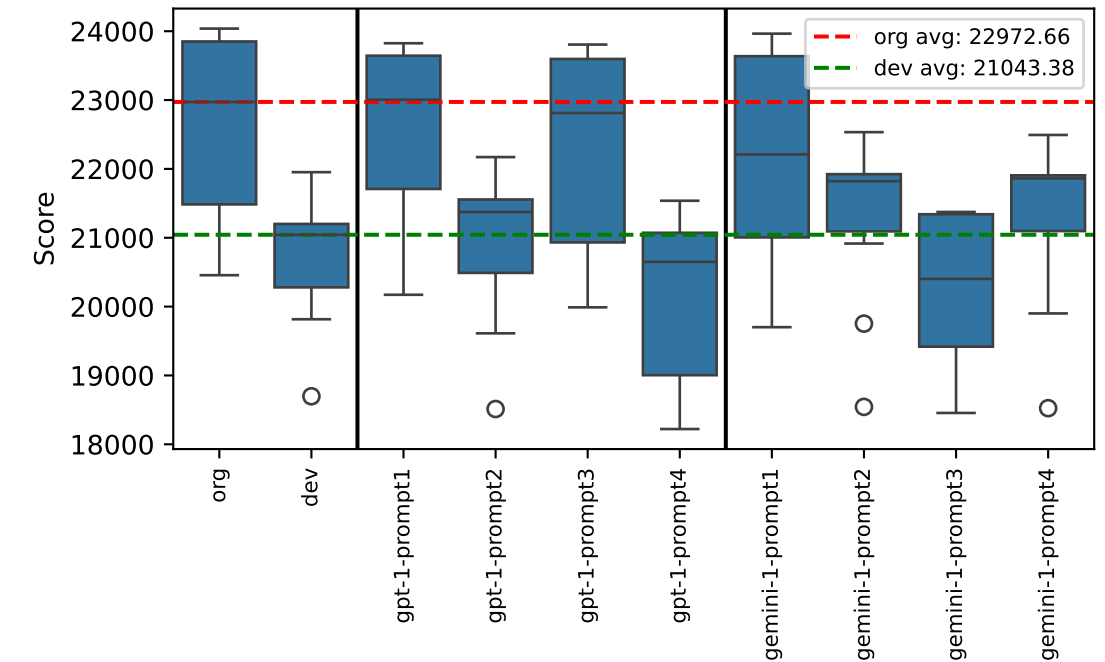
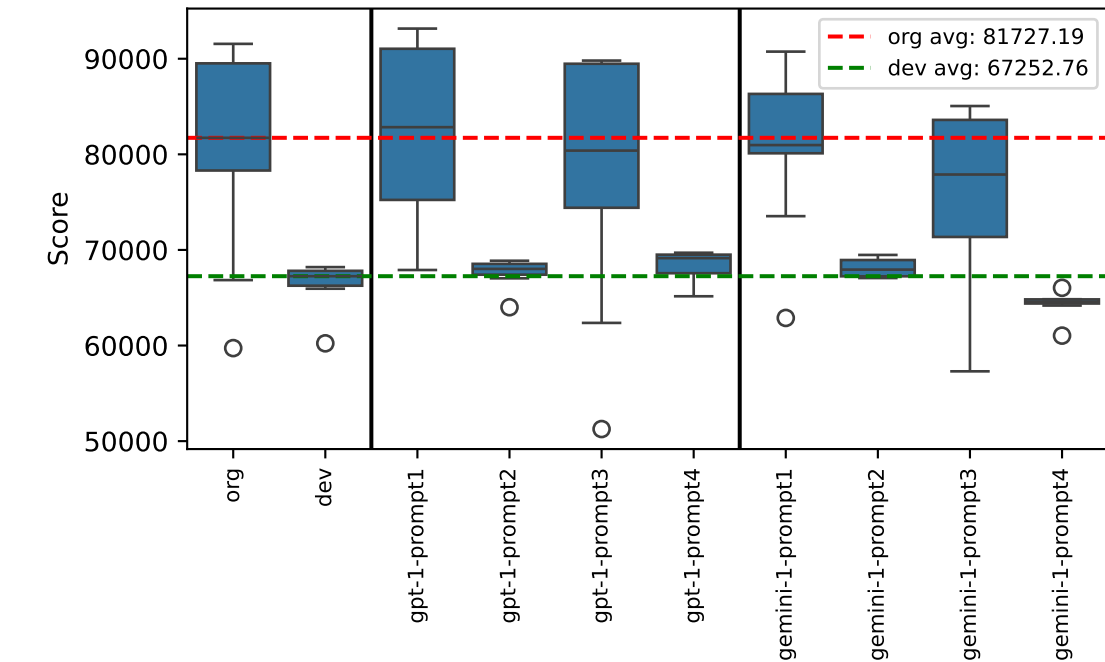


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '100', 'noCaching': 'true', 'supplier': 'org.apache.kafka.common.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign', 'version': '2.0.0', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '100', 'noCaching': 'true', 'supplier': 'org.apache.kafka.common.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign', 'version': '2.0.0'

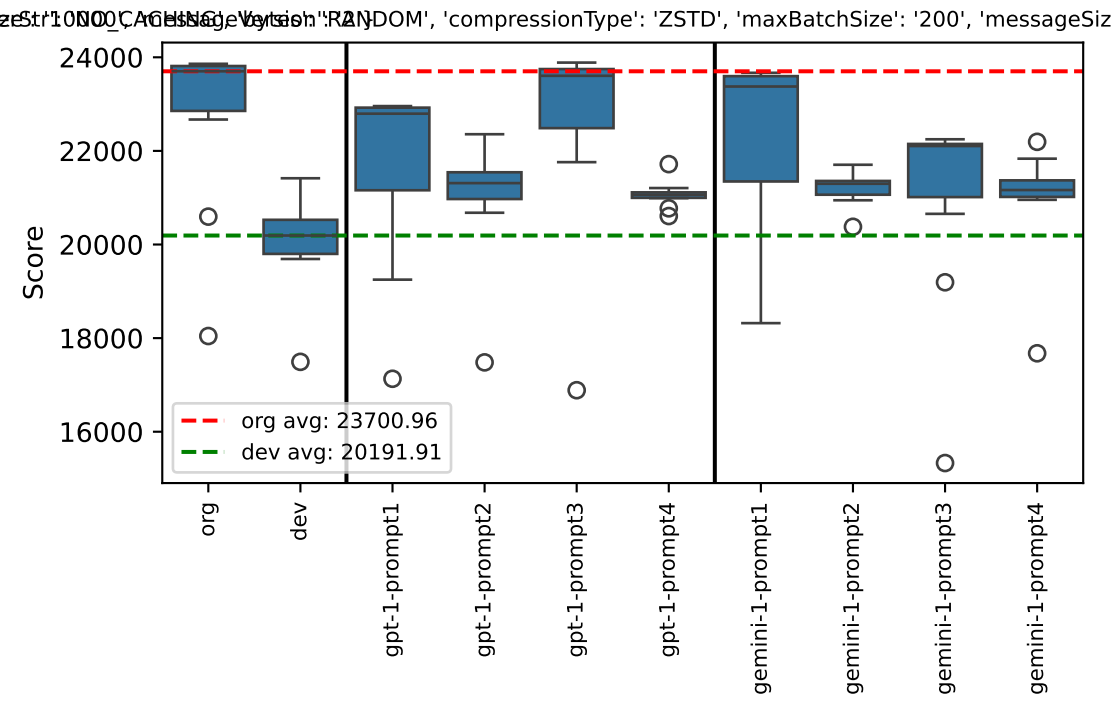
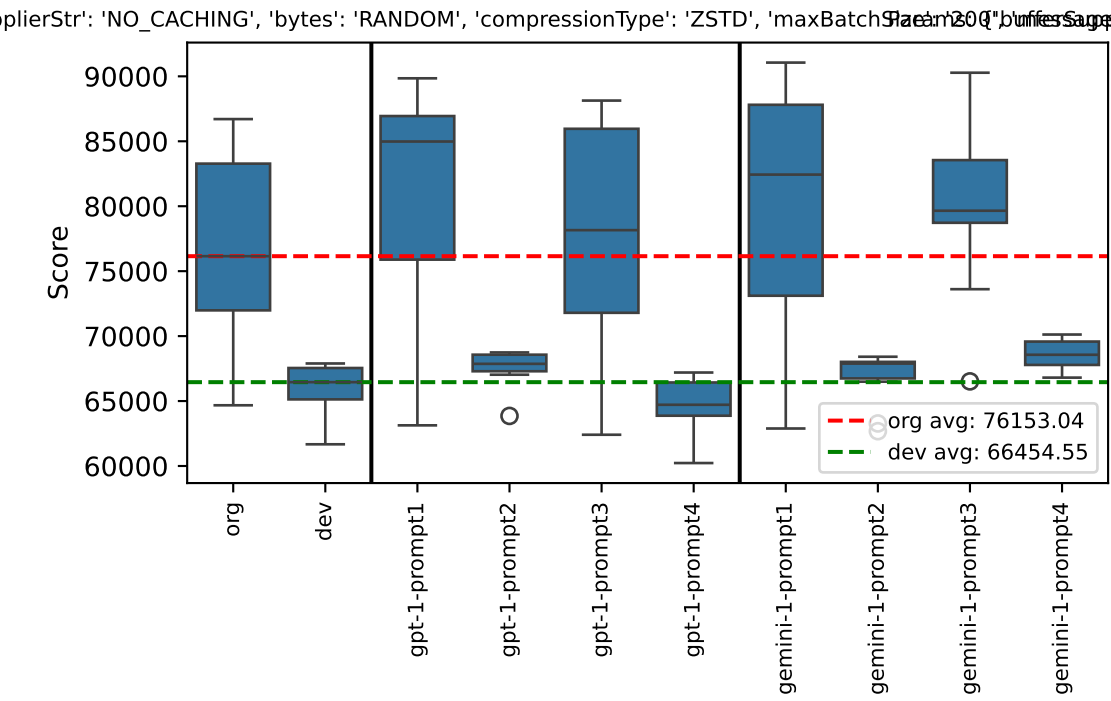
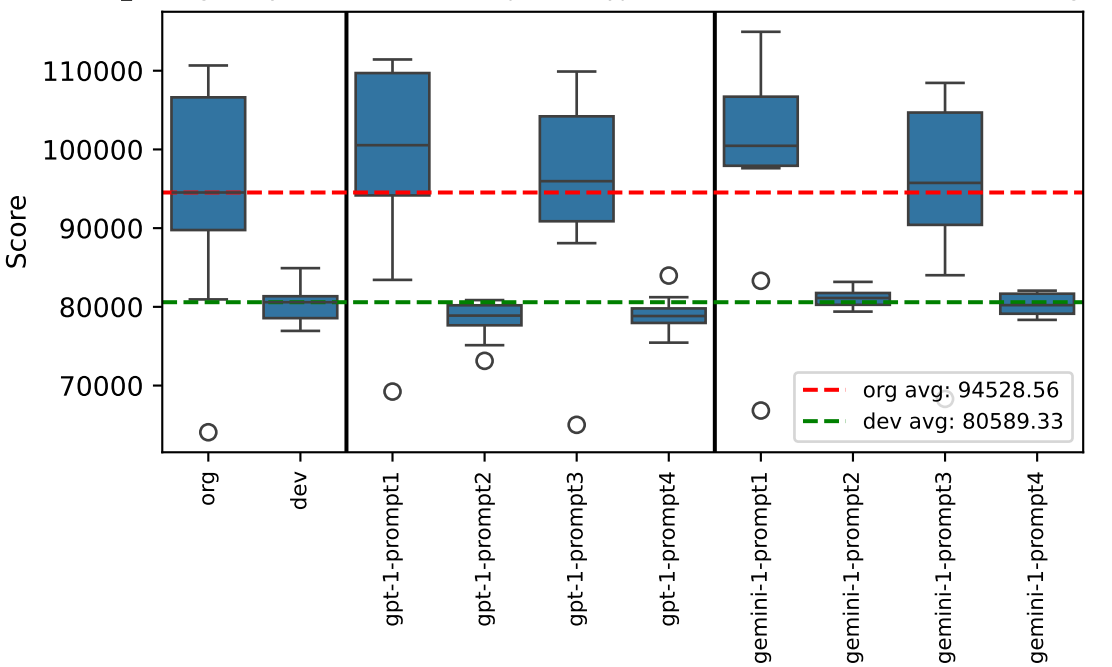
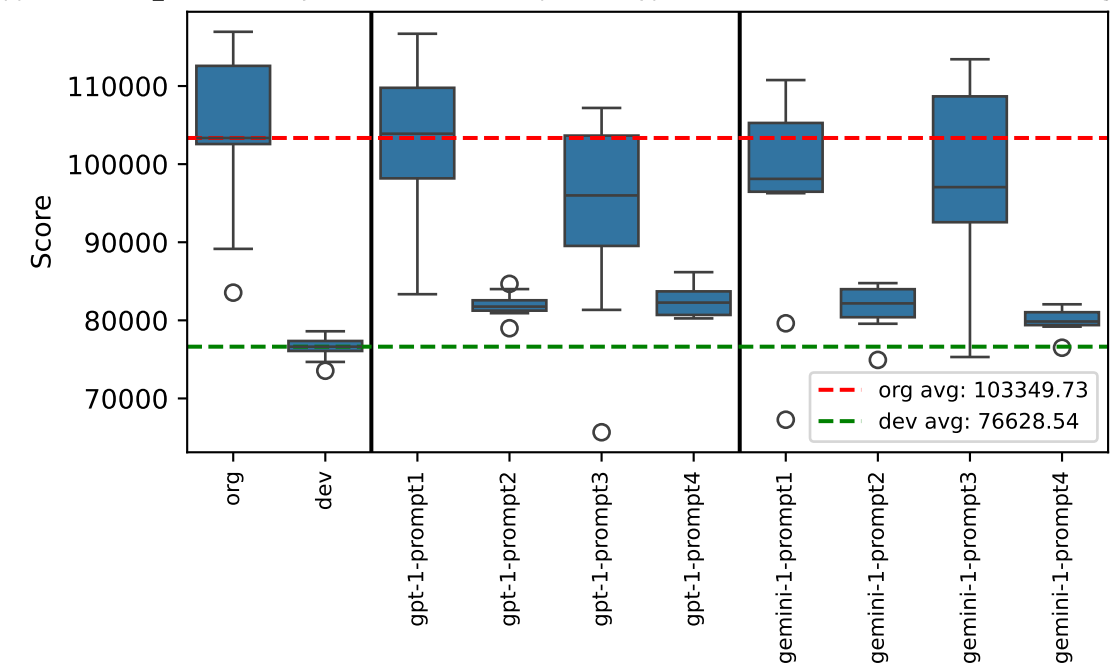


SupplierStr: 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '100', 'noCaching': 'true', 'supplier': 'org.apache.kafka.common.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign', 'version': '2.0.0', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '100', 'noCaching': 'true', 'supplier': 'org.apache.kafka.common.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign', 'version': '2.0.0'



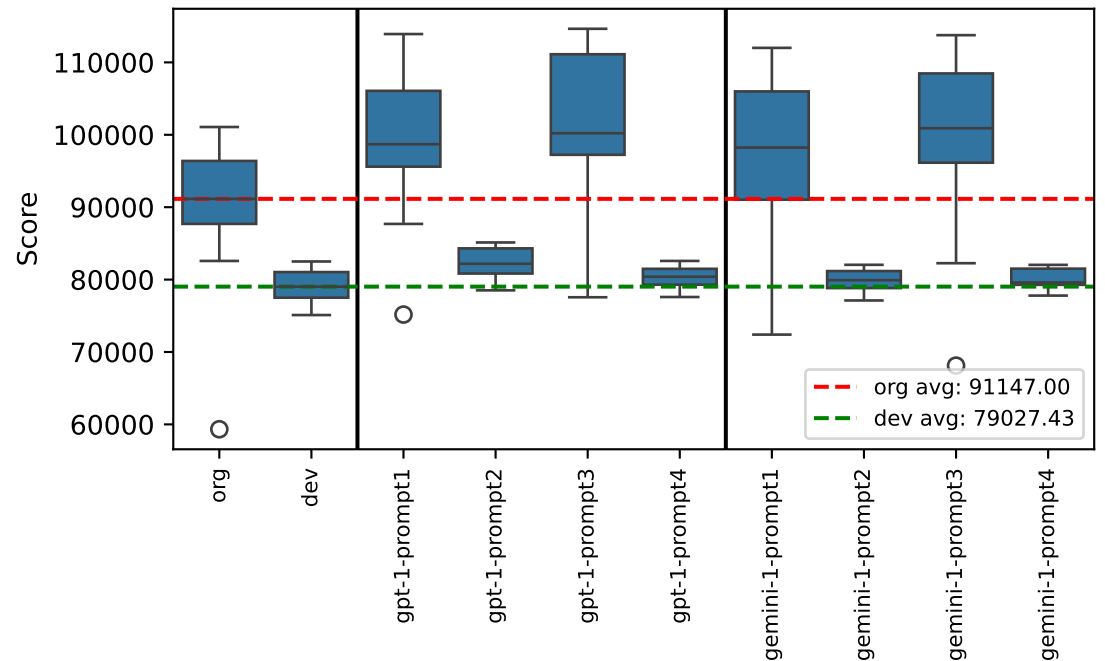
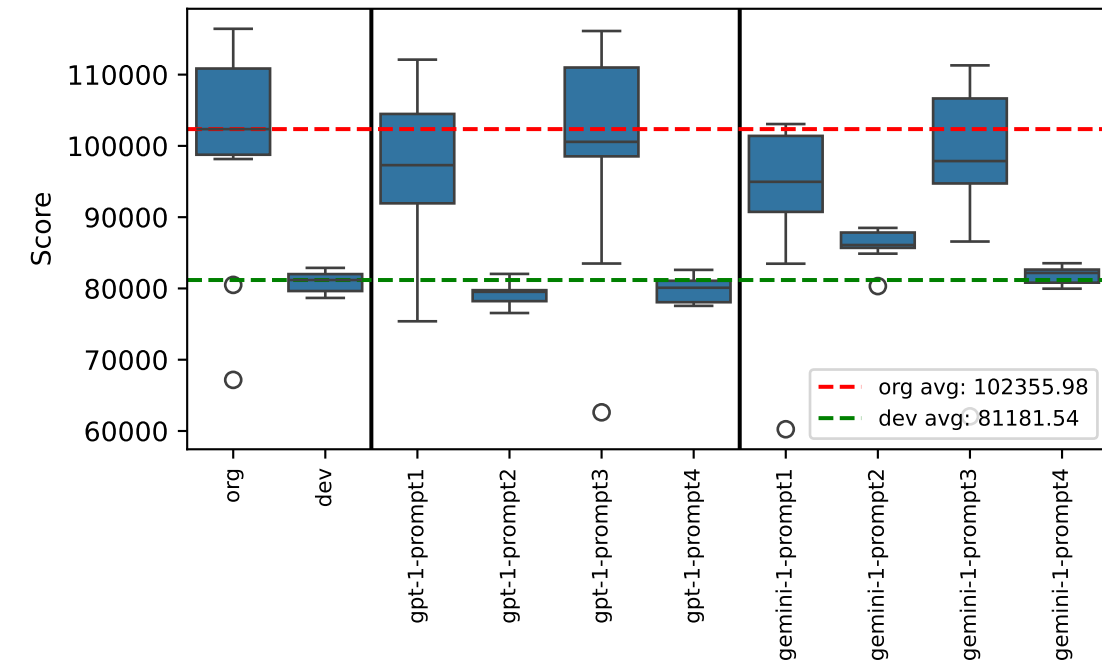
che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize': '100', 'numSuppliers': '5', 'numVersions': '2', 'orgVersion': '2', 'supplierStr': 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize': '100', 'numSuppliers': '5', 'numVersions': '2', 'orgVersion': '2'

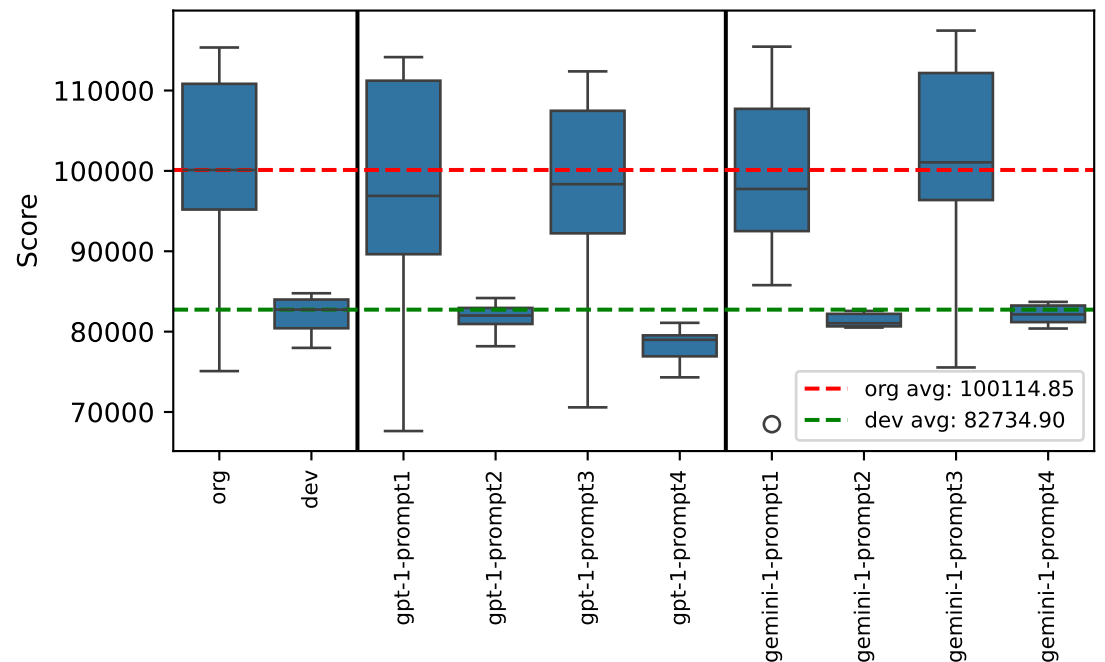
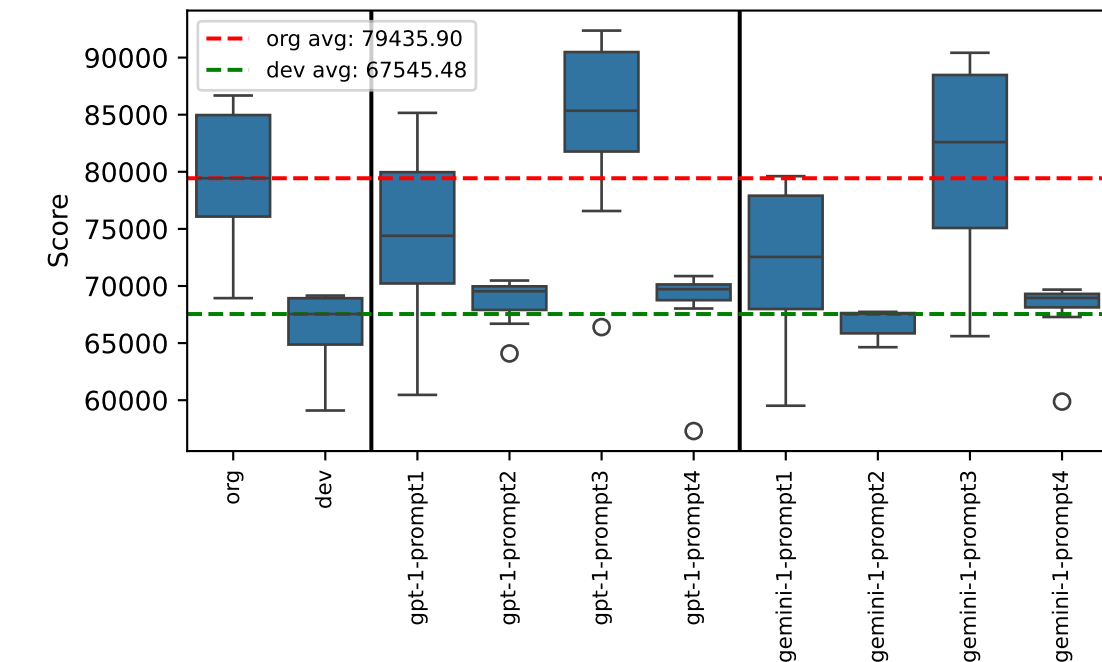


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '100', 'numSuppliers': '2', 'orgVersion': '2', 'promptVersion': '2', 'supplierStr': 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '100', 'numSuppliers': '2', 'orgVersion': '2', 'promptVersion': '2'

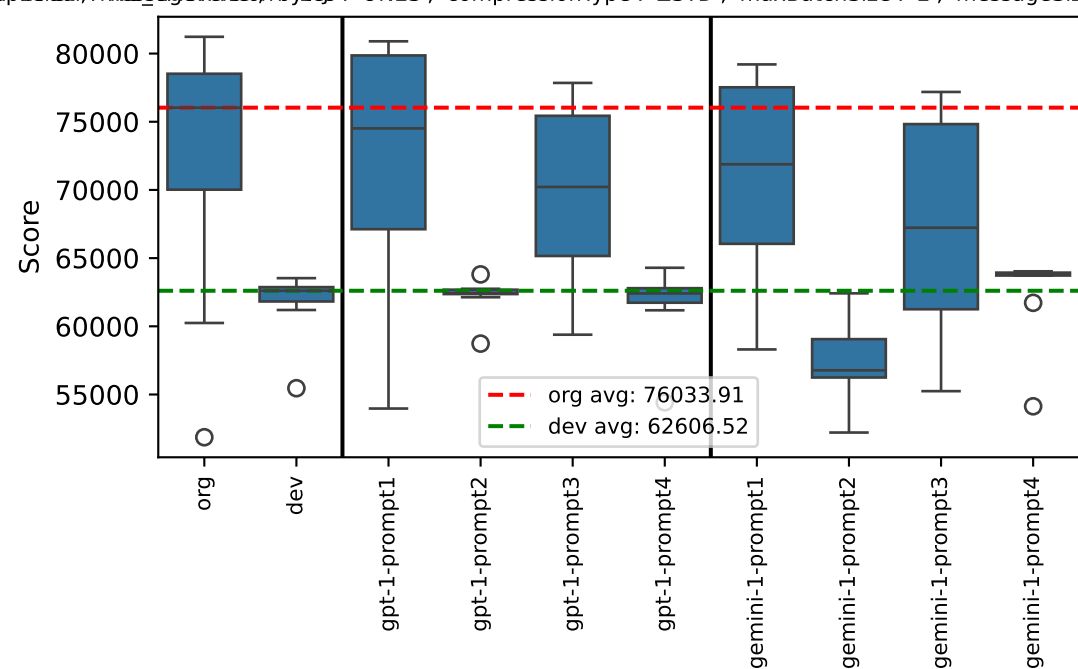
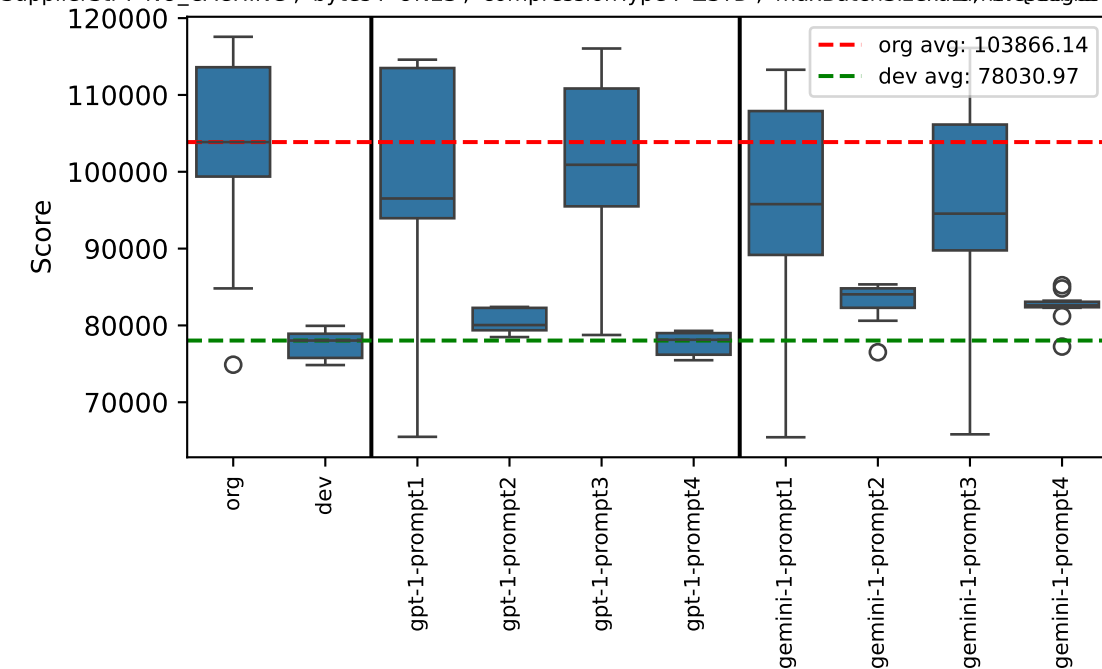


SupplierStr: 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '100', 'numSuppliers': '2', 'orgVersion': '2', 'promptVersion': '2', 'supplierStr': 'NO_CACHING', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '100', 'numSuppliers': '2', 'orgVersion': '2', 'promptVersion': '2'

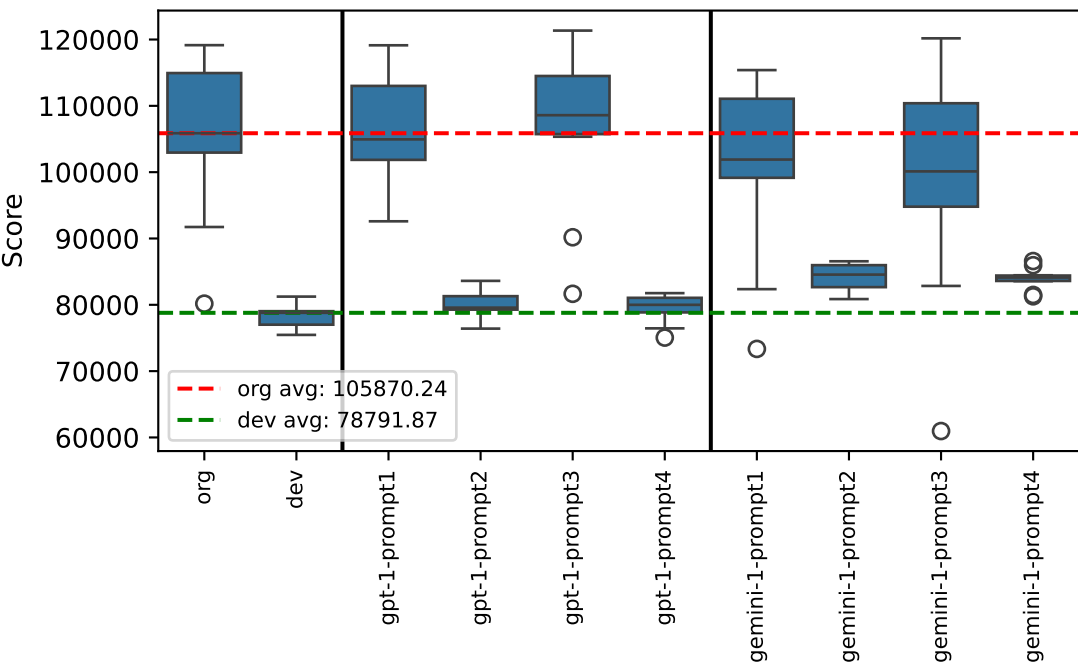
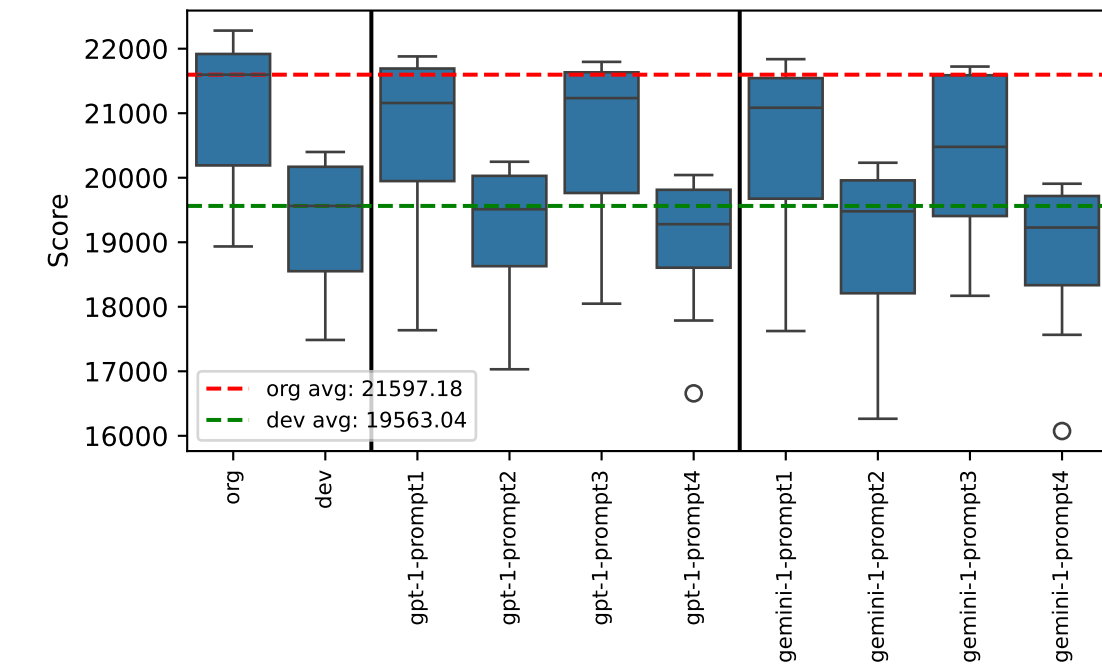


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': 12, 'messageSize': 1000, 'noCache': 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize':

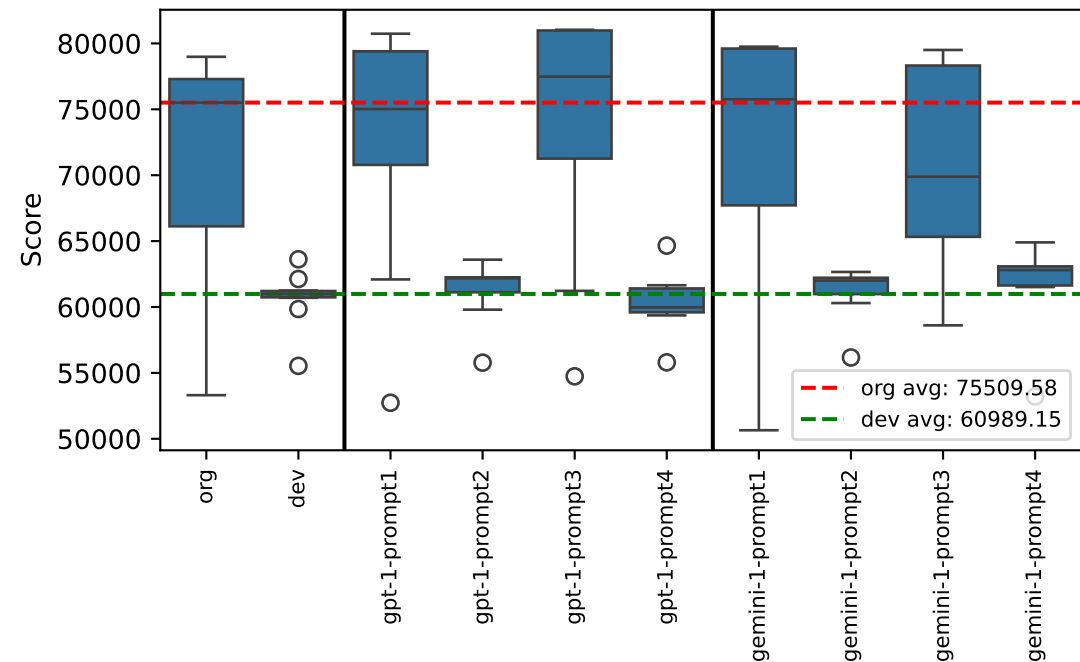
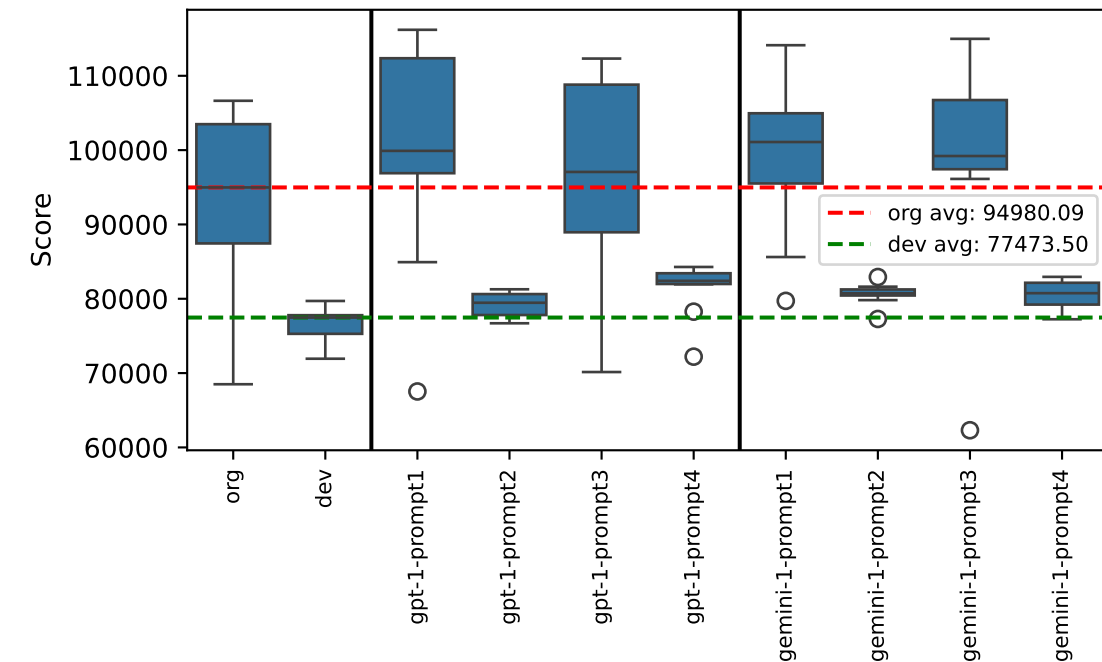


SupplierStr: 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': 2, 'messageSize': 500, 'noCache': 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize':

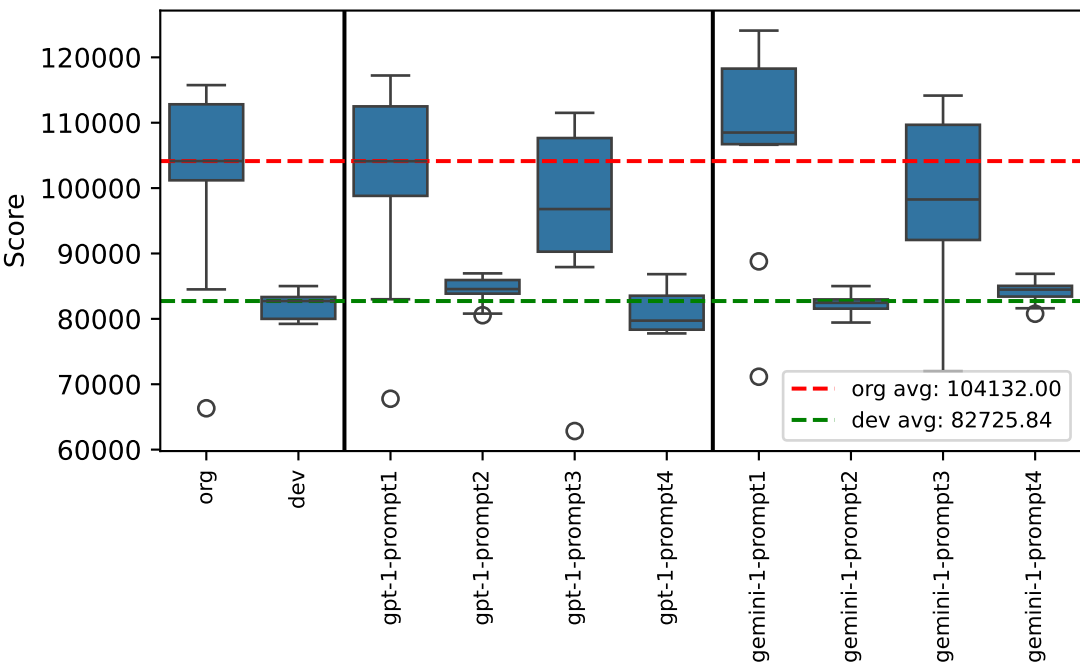
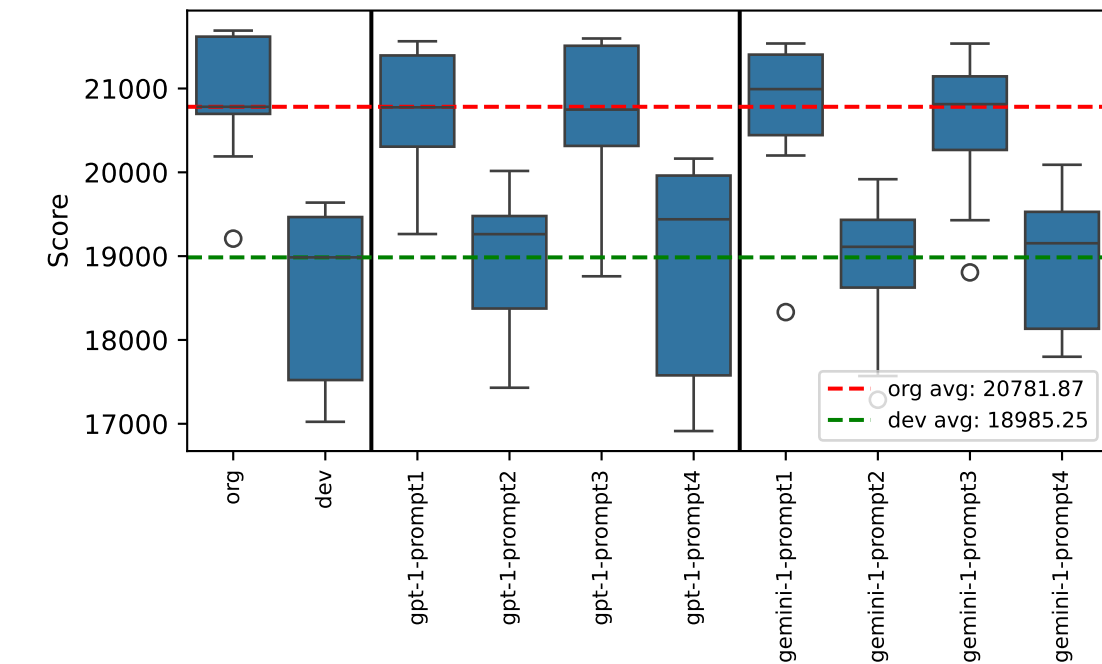


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

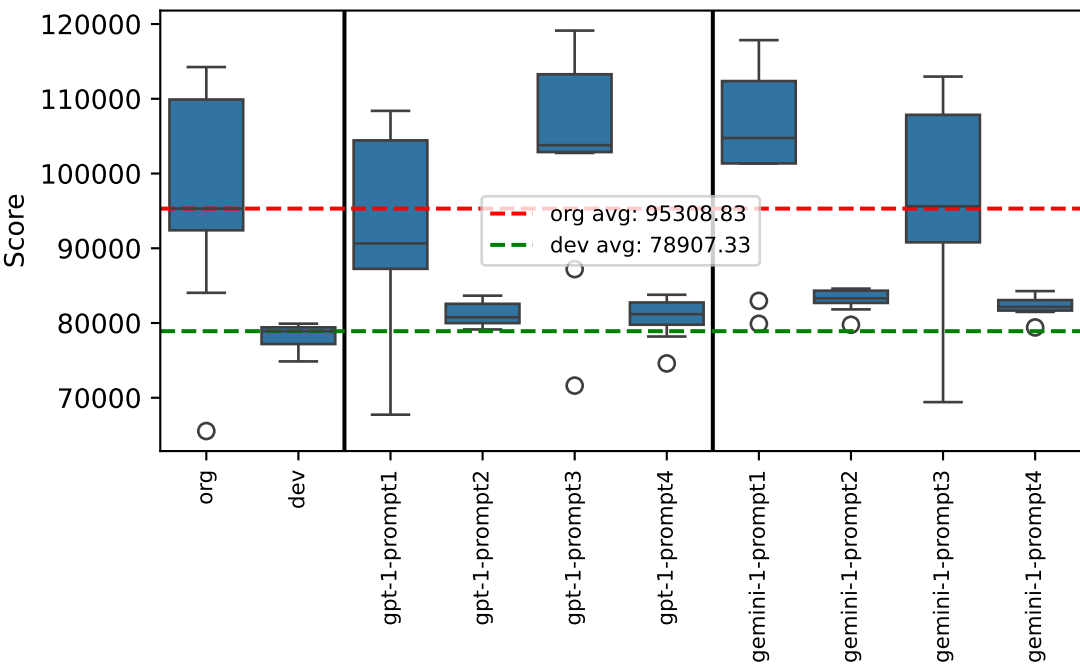
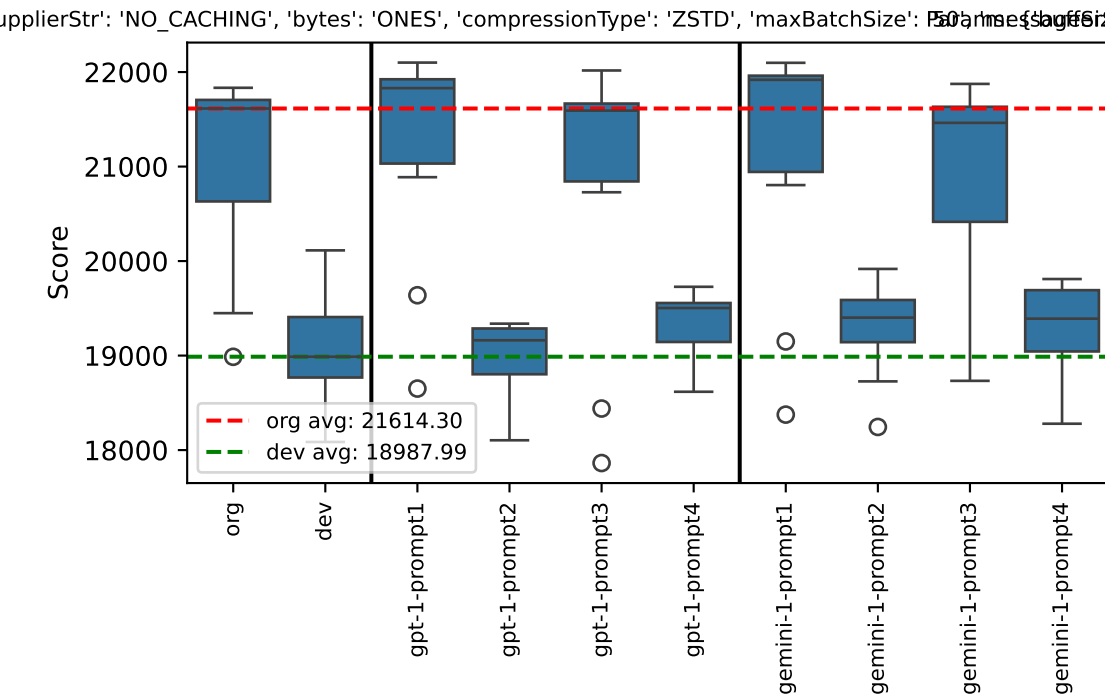
SupplierStr: 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '1000', 'No_Caching': 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '1000'



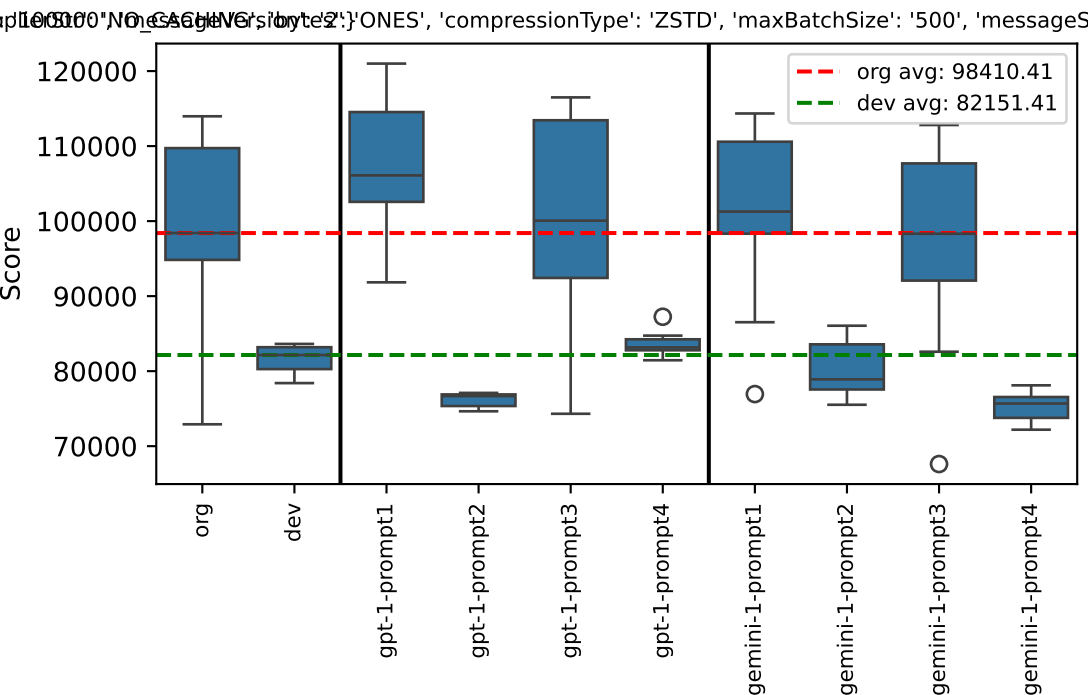
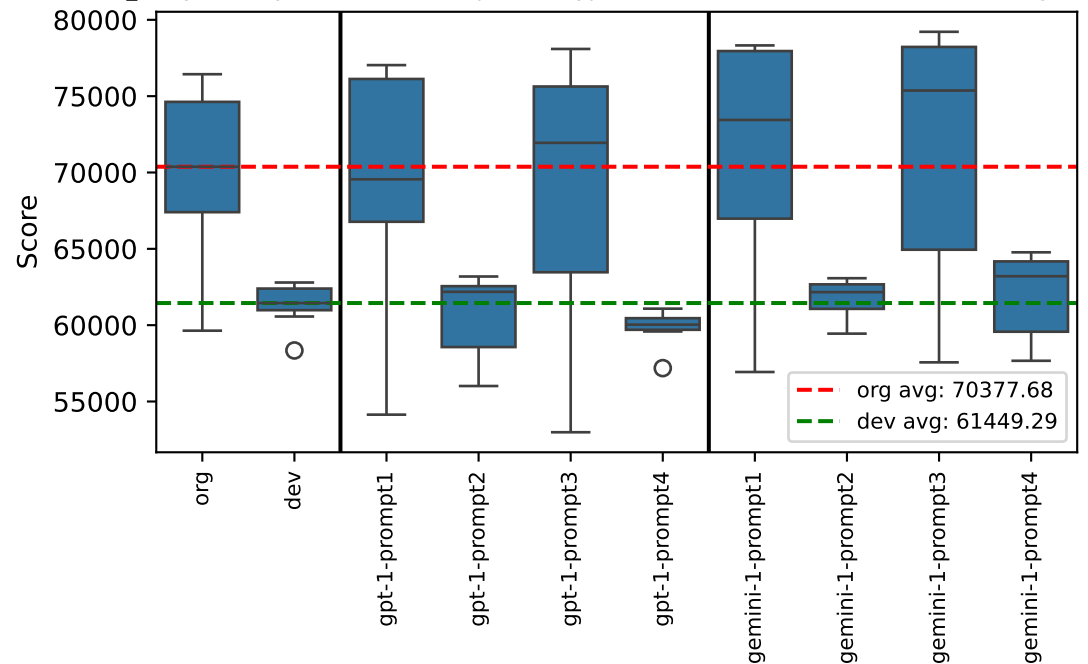
SupplierStr: 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '1000', 'No_Caching': 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize': '1000'



```
SupplierStr: 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize': '100', 'noCaching': 'NO_CACHING', 'bytes2': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize':
```

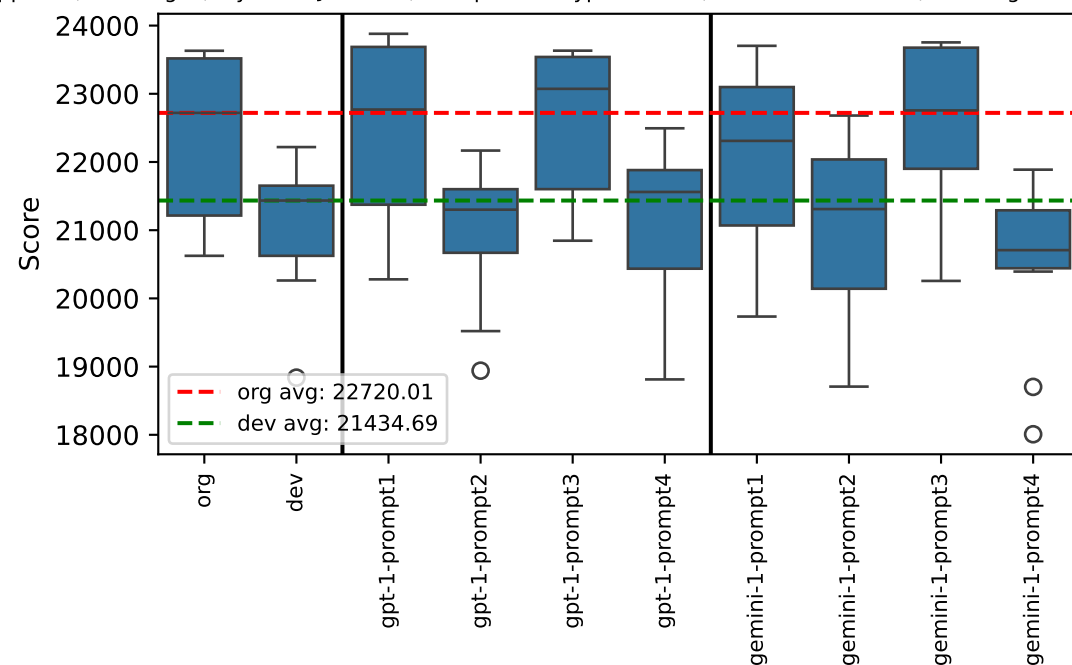
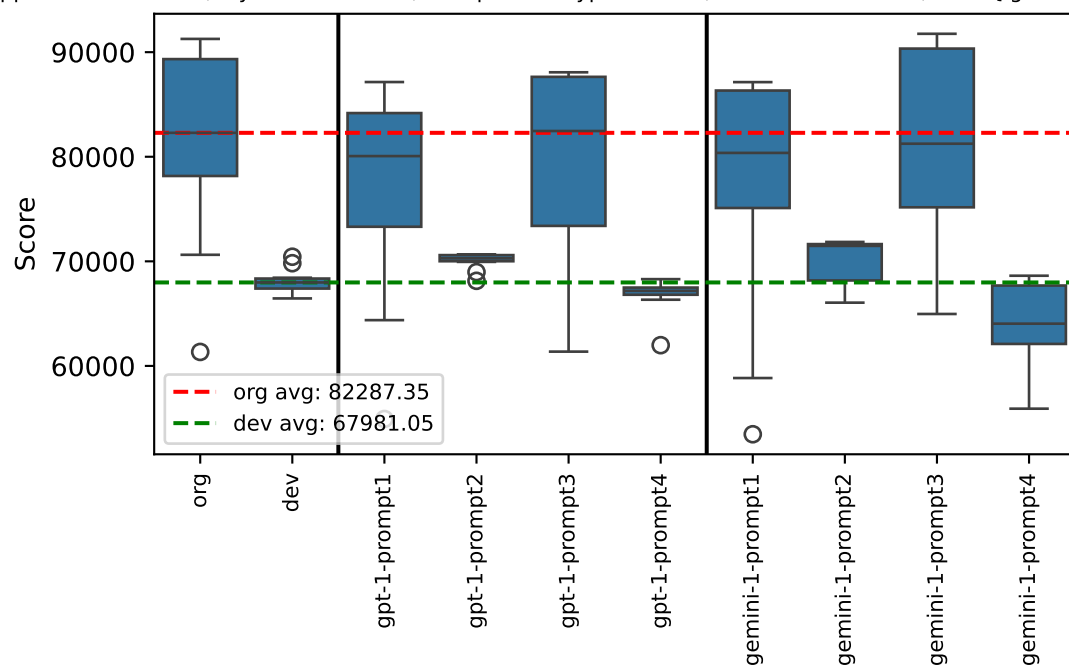


```
supplierStr': 'NO_CACHING', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize': '100', 'noCacheVersion': 'byte2', 'ones': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize'
```

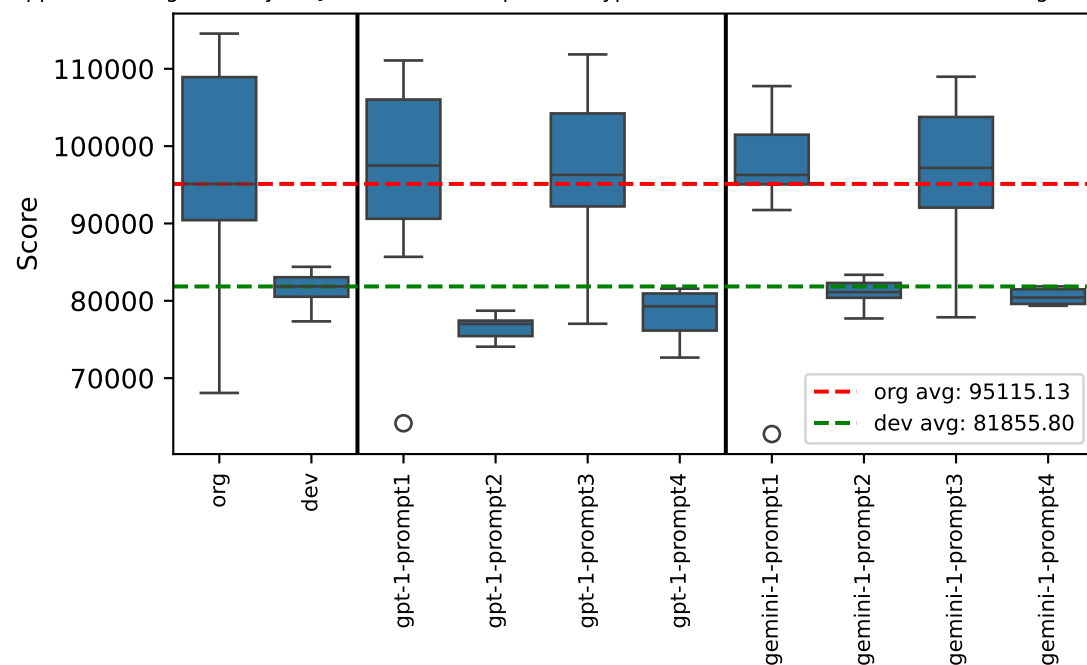
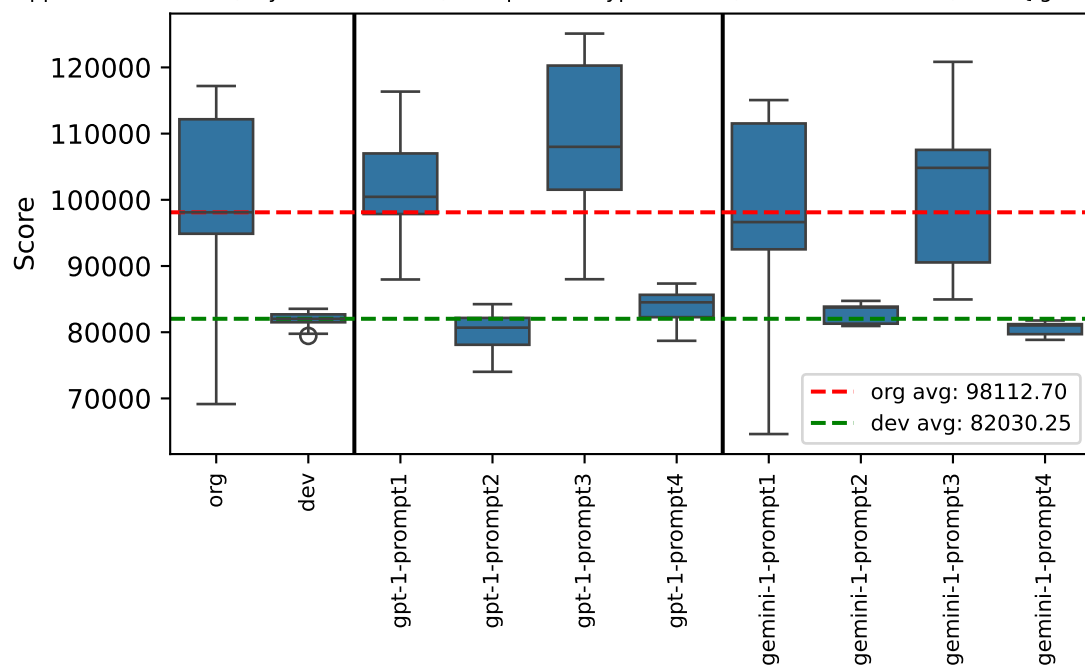


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

```
'SupplierStr': 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '100000', 'messageSize': '100000', 'parameters': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '1'
```

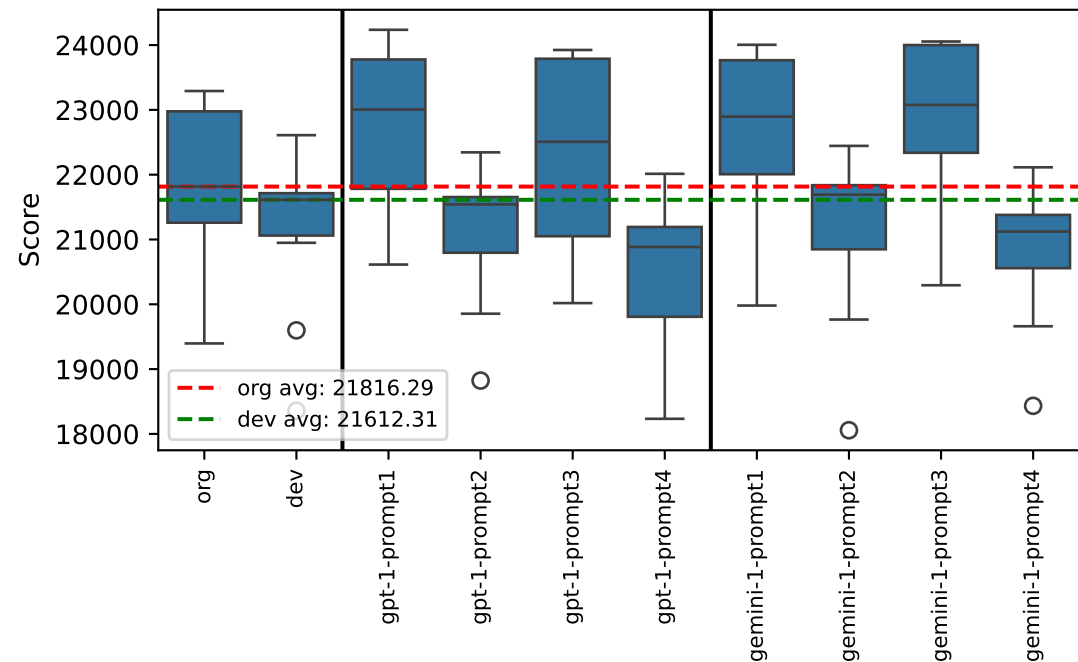
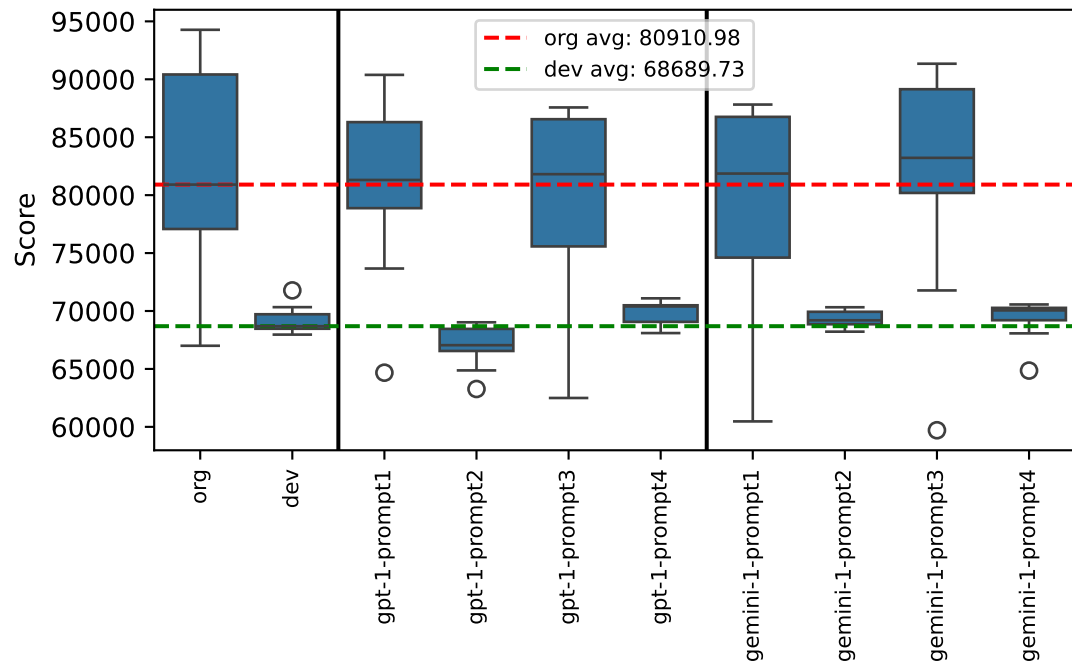


```
erSupplierStr': 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': '100', 'messageVersion': '2.3', 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize':
```

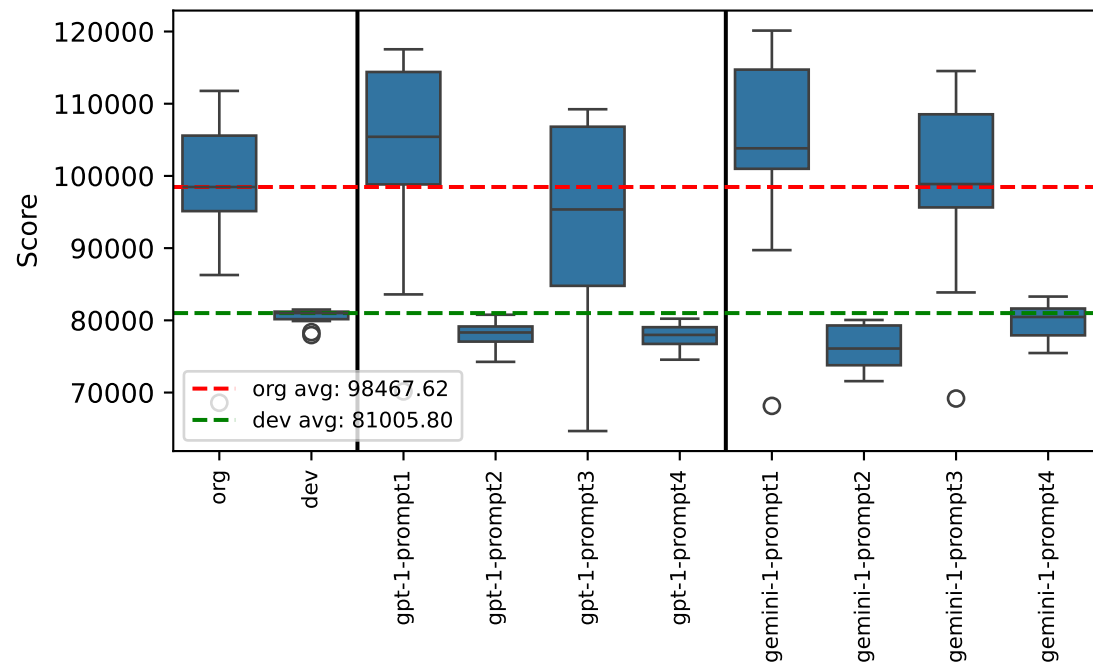
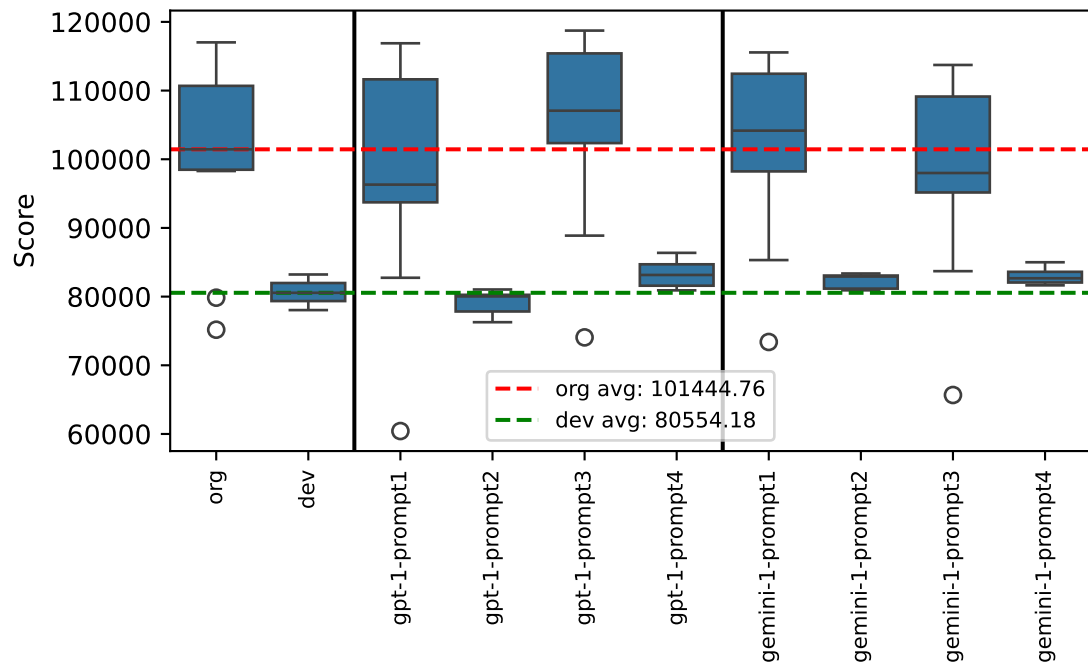


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': '1000', 'messageVersion': '2', 'messageBytes': '2048', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': '1000', 'messageVersion': '2', 'messageBytes': '2048'

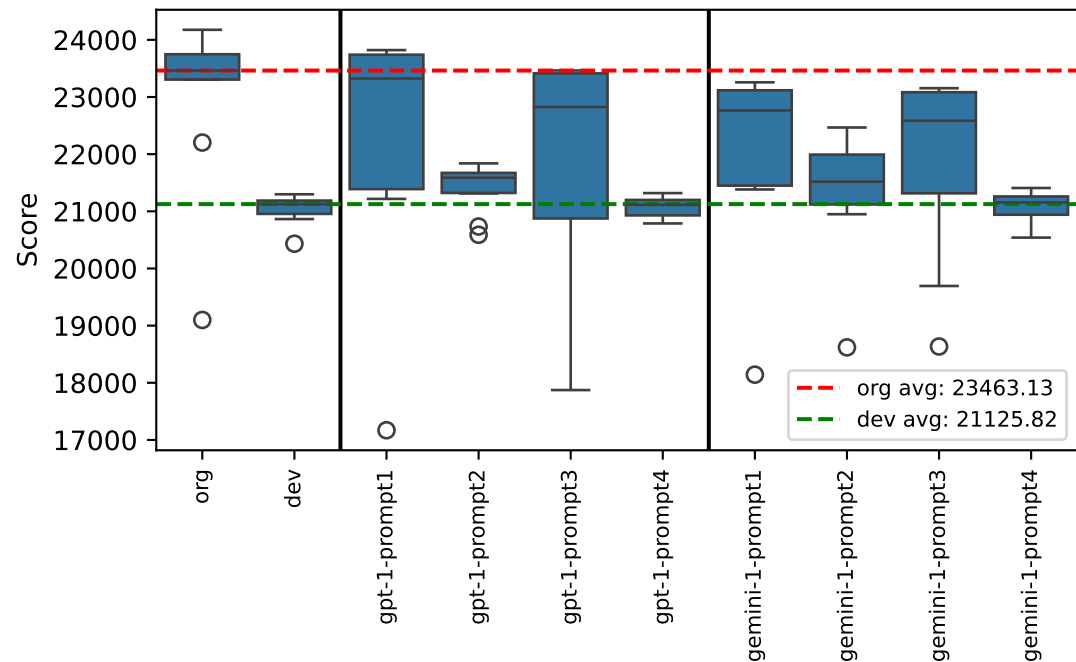
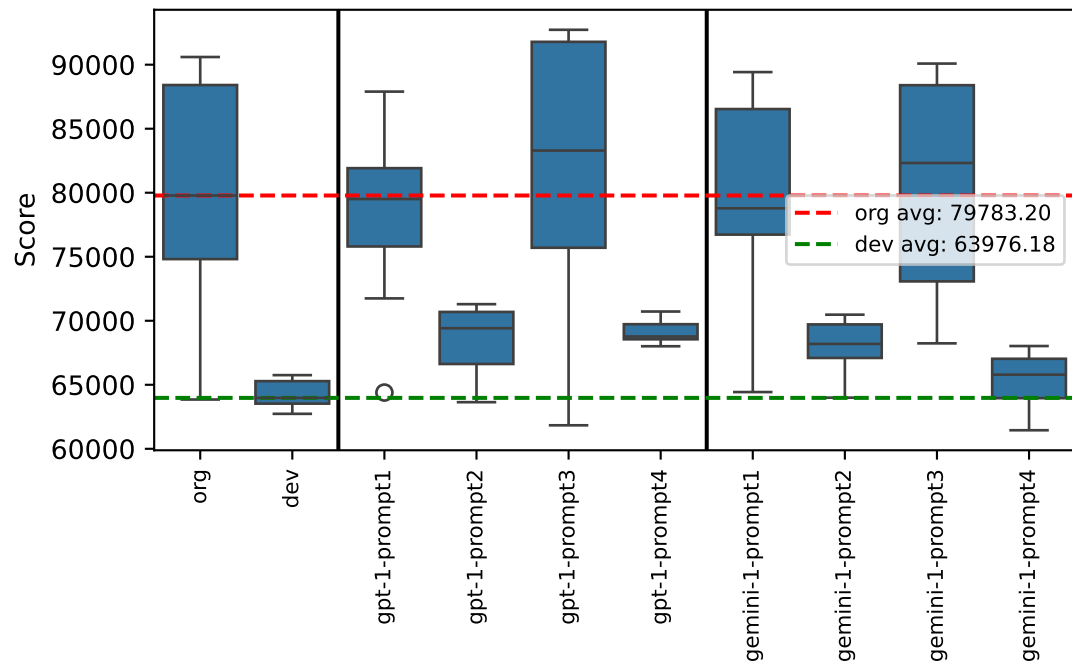


SupplierStr: 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '1000', 'messageVersion': '2', 'messageBytes': '2048', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '1000', 'messageVersion': '2', 'messageBytes': '2048'

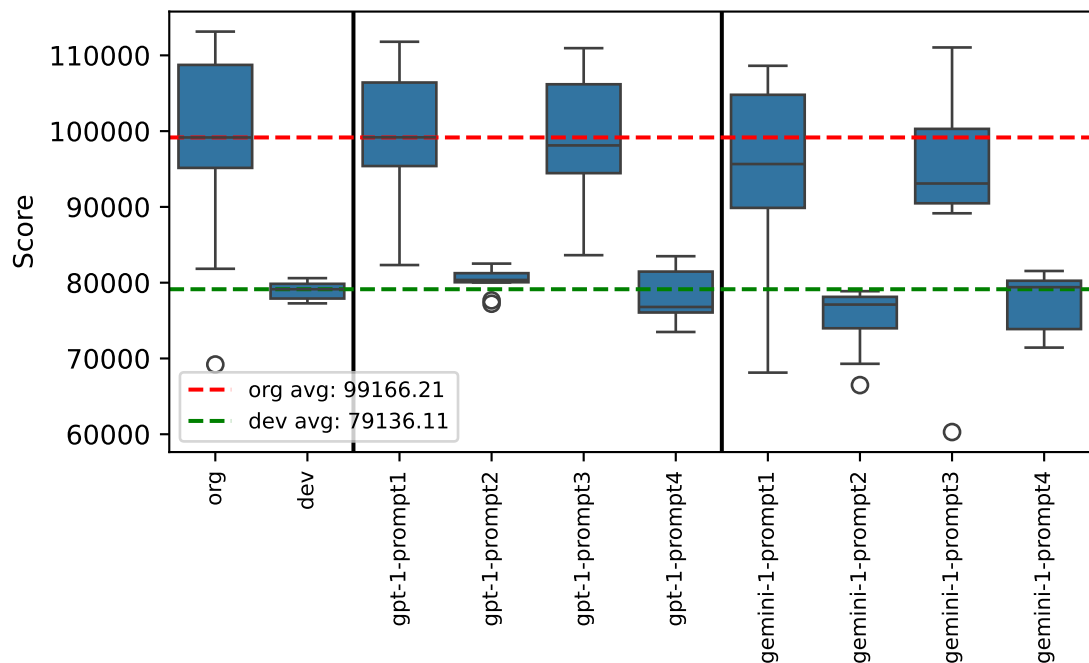
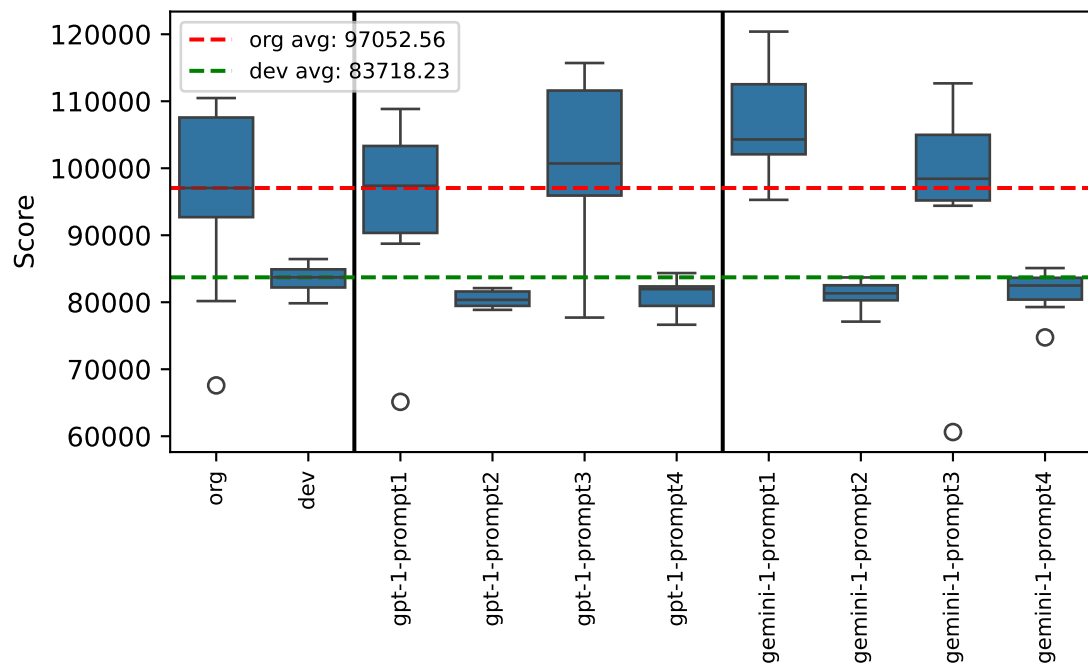


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize': '1000', 'messageVersion': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '200', 'messageSize':

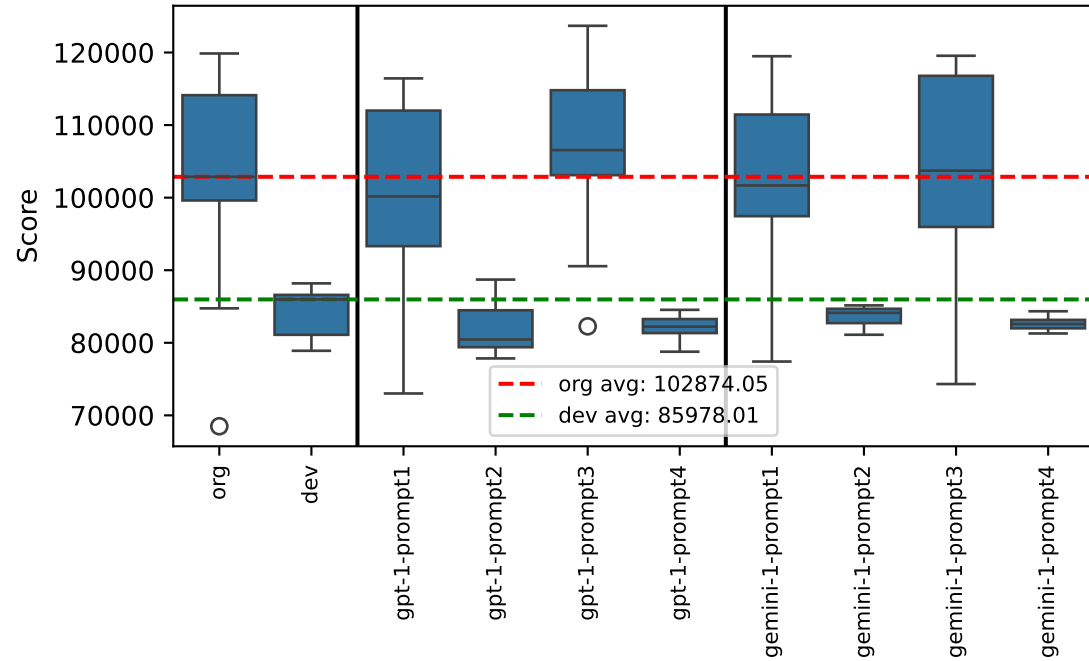
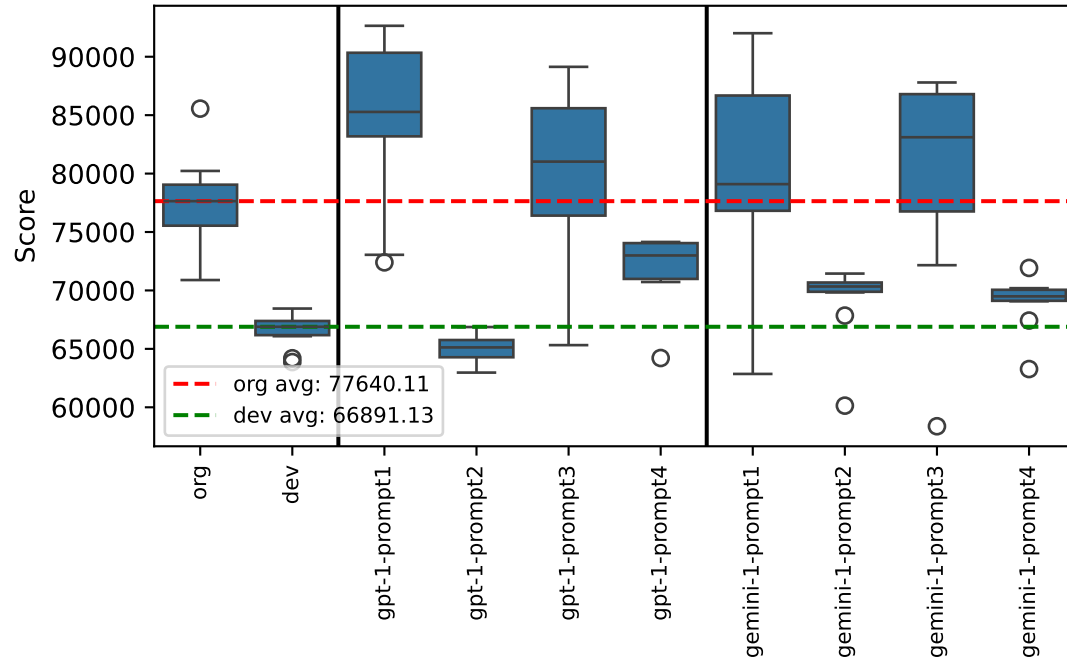


SupplierStr: 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '1000', 'messageVersion': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize':

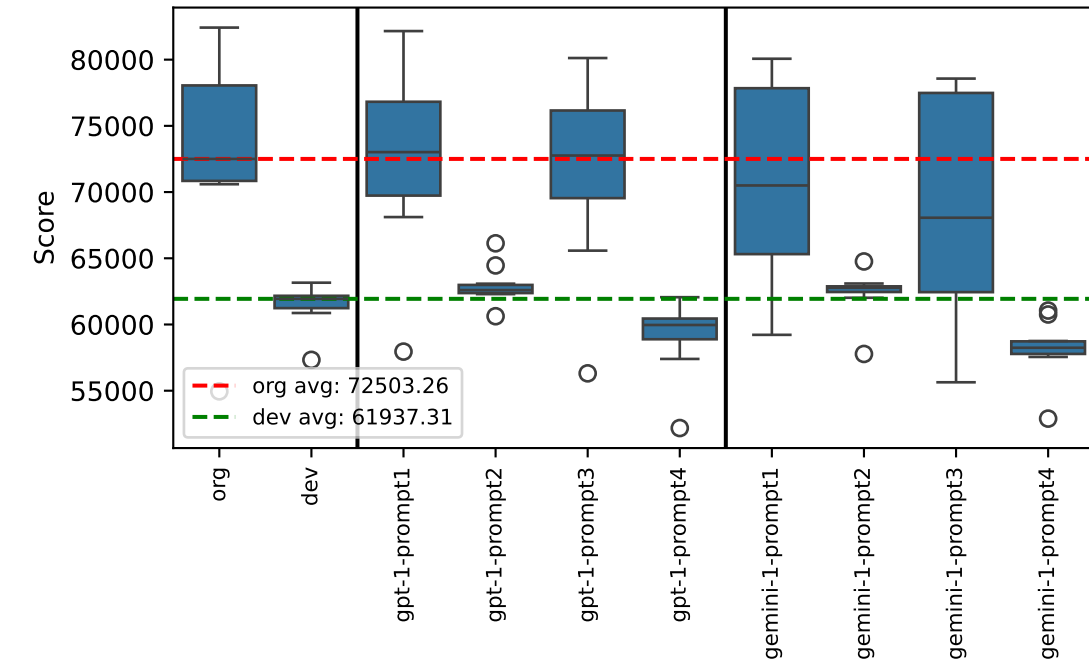
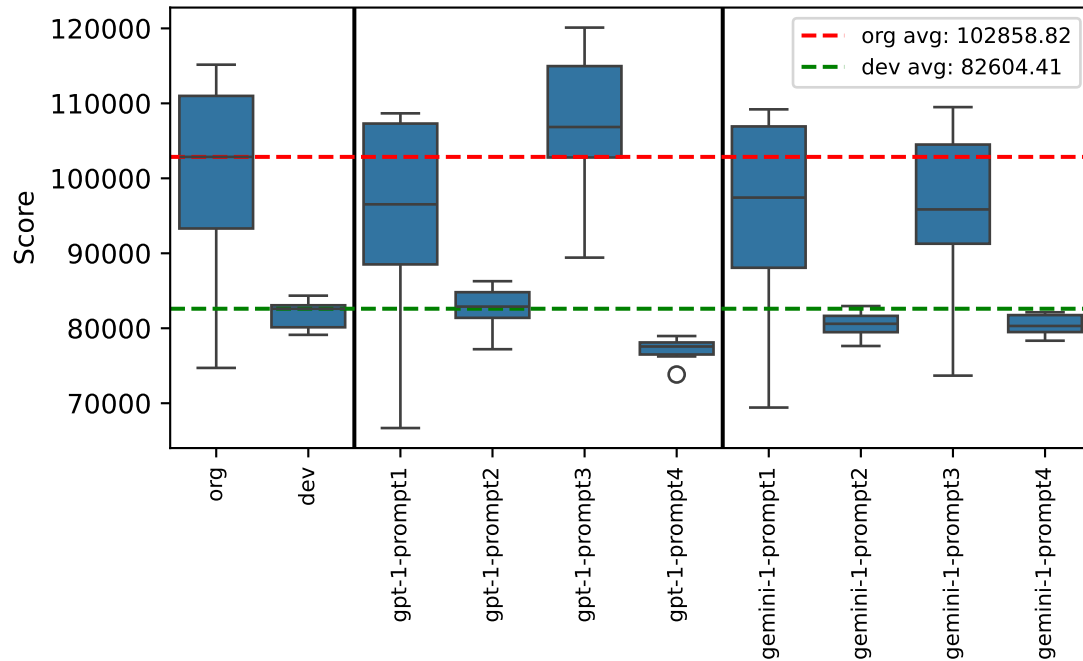


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

SupplierStr: 'CREATE', 'bytes': 'RANDOM', 'compressionType': 'ZSTD', 'maxBatchSize': '500', 'messageSize': '5000', 'messageSupplier': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '1'

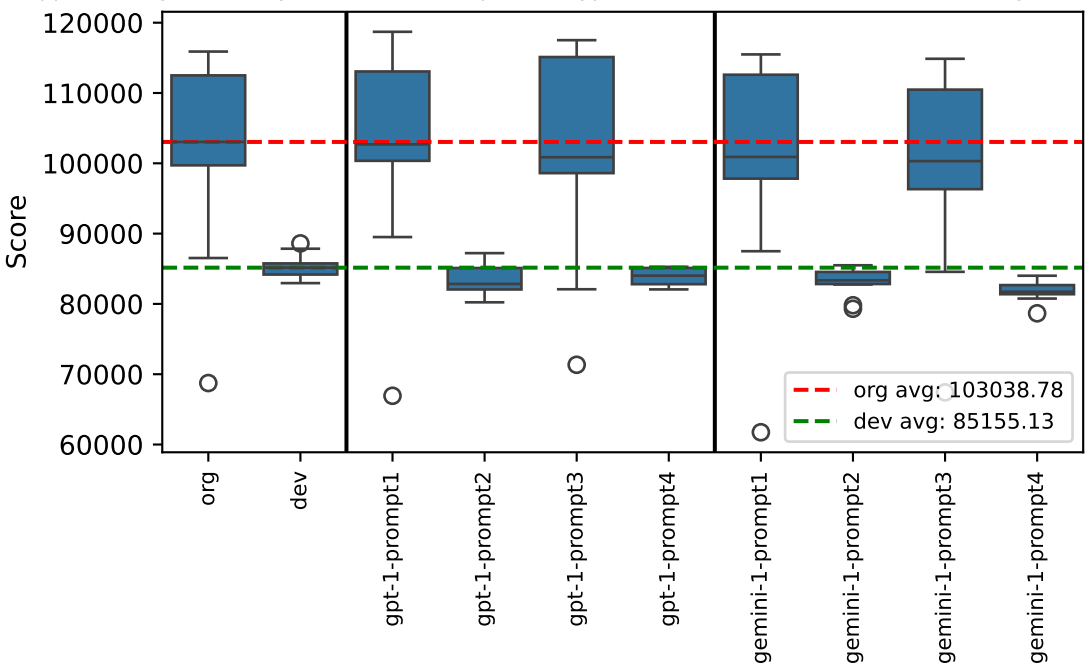
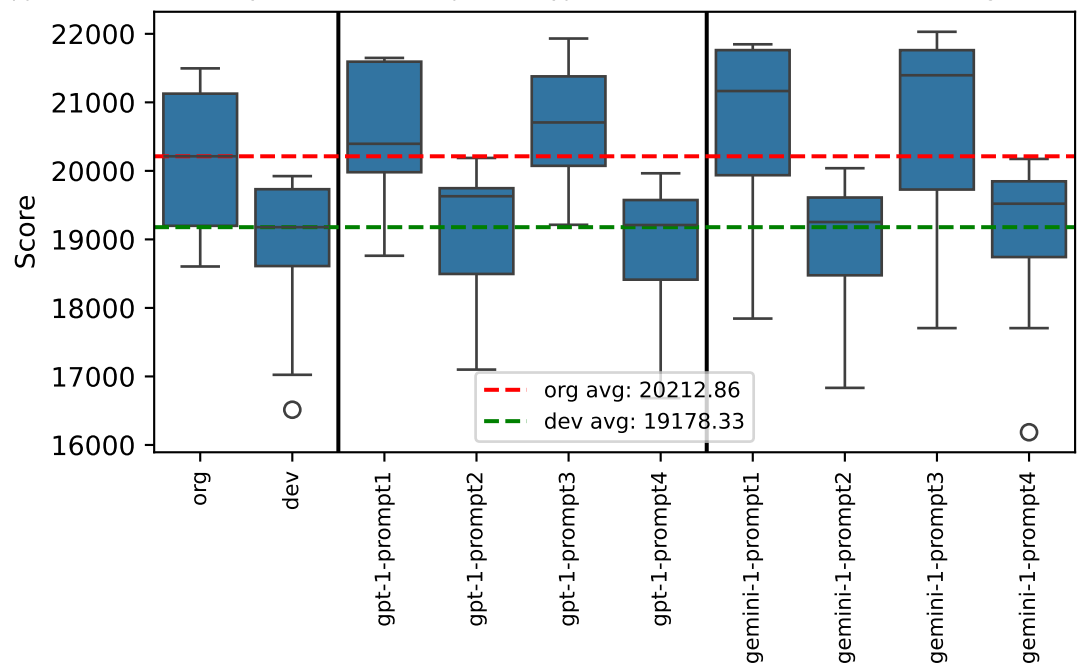


SupplierStr: 'CREATE', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '500', 'messageSupplier': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': '1'

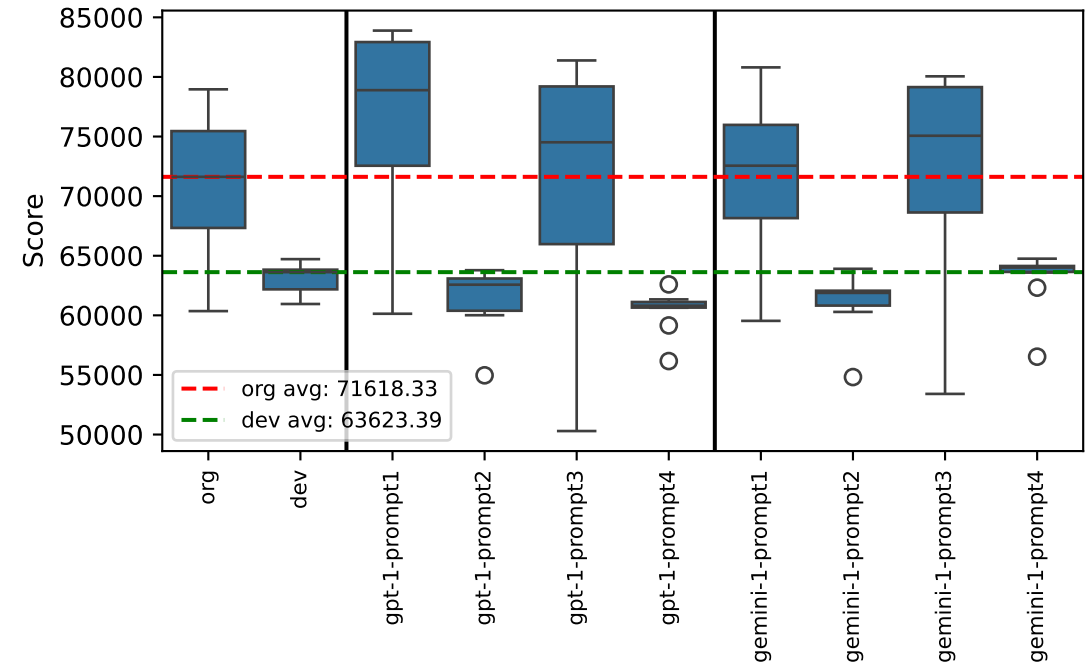
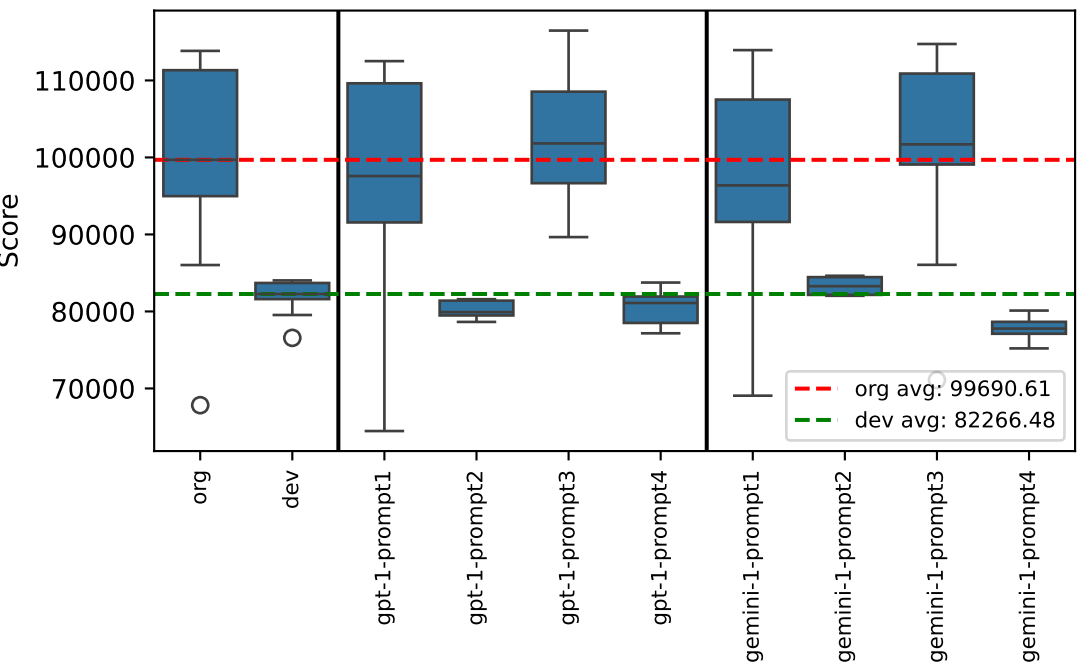


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

erSupplierStr: 'CREATE', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '1', 'messageSize': 1000, 'numSuppliers': 10, 'numValidators': 10, 'numWorkers': 10, 'offset': 0, 'offsetBytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': '1'

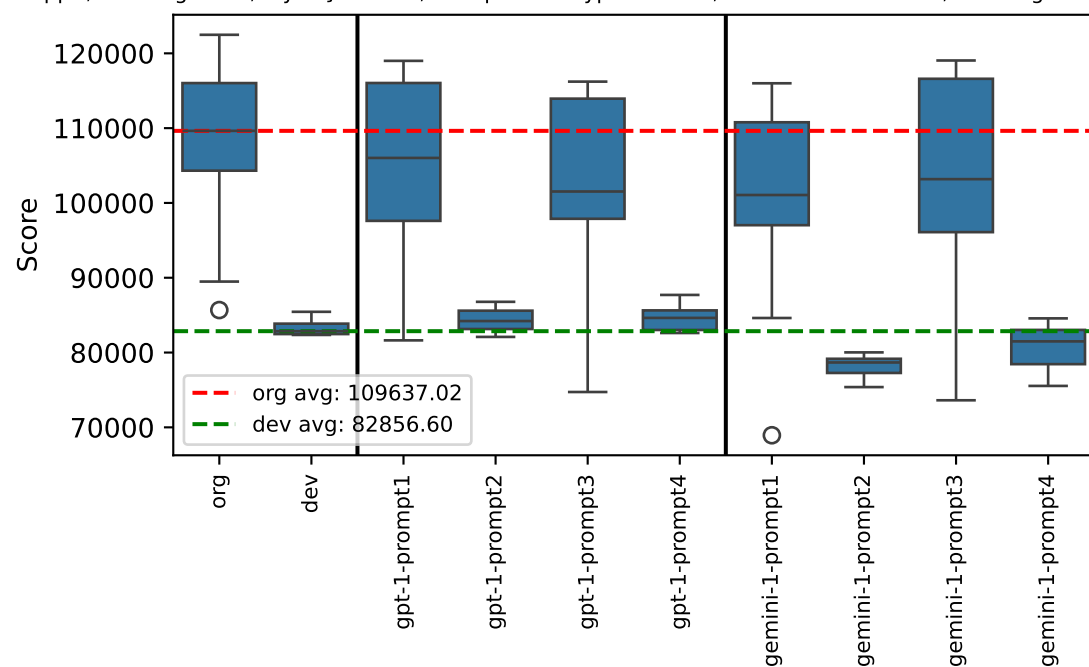
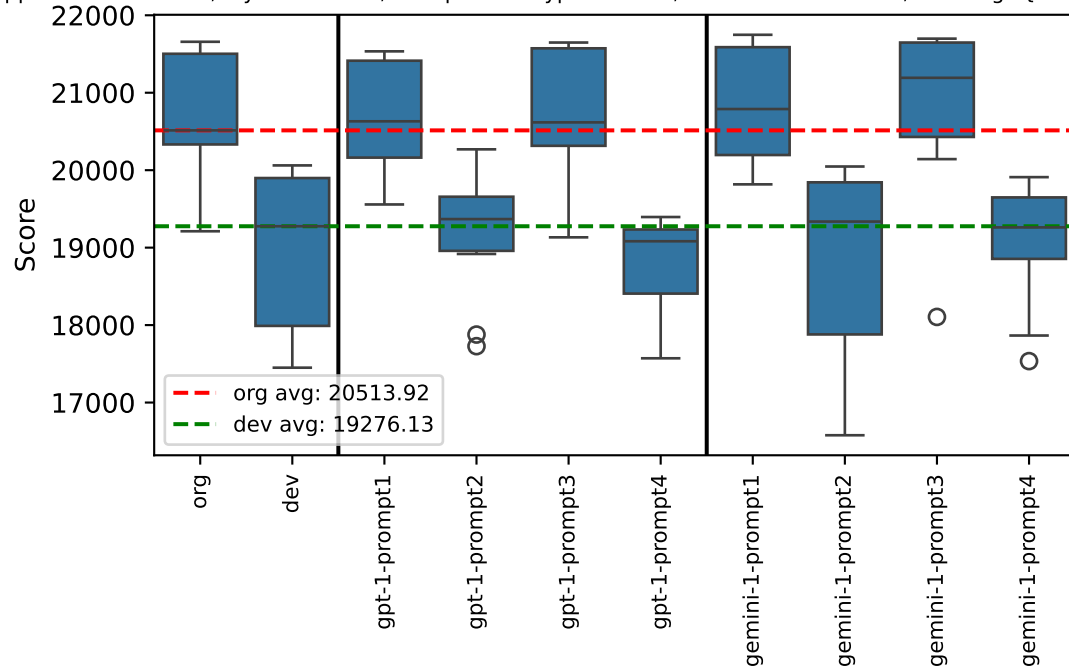


erSupplierStr: 'CREATE', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': 1000, 'numSuppliers': 10, 'numValidators': 10, 'numWorkers': 10, 'offset': 0, 'offsetBytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '2', 'messageSize': '1'

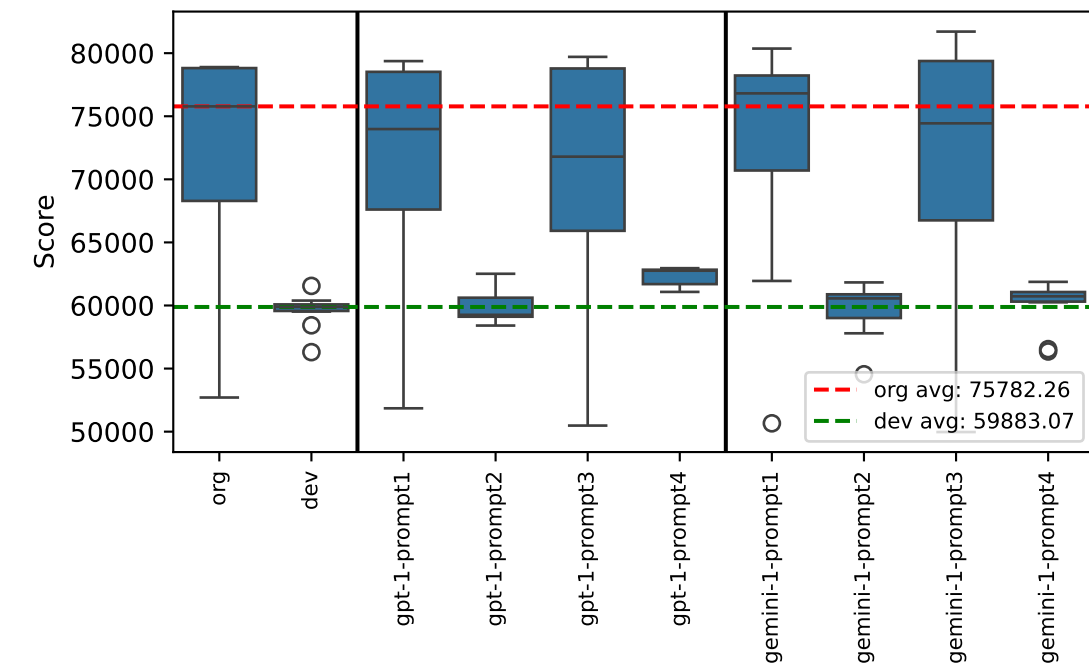
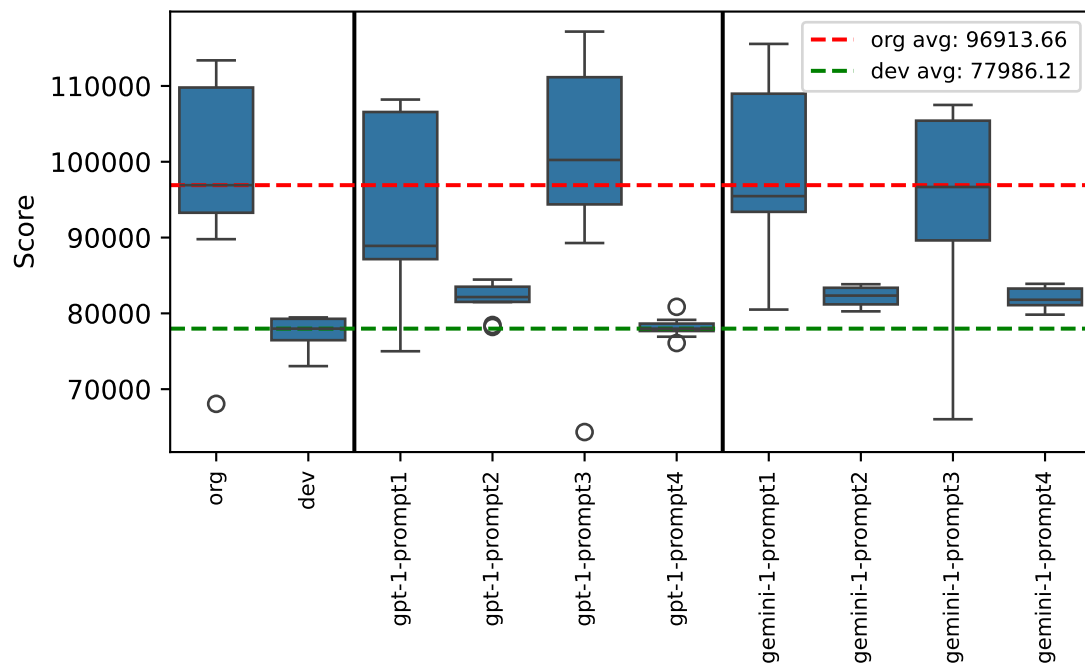


che.kafka.jmh.record.CompressedRecordBatchValidationBenchmark.measureValidateMessagesAndAssign

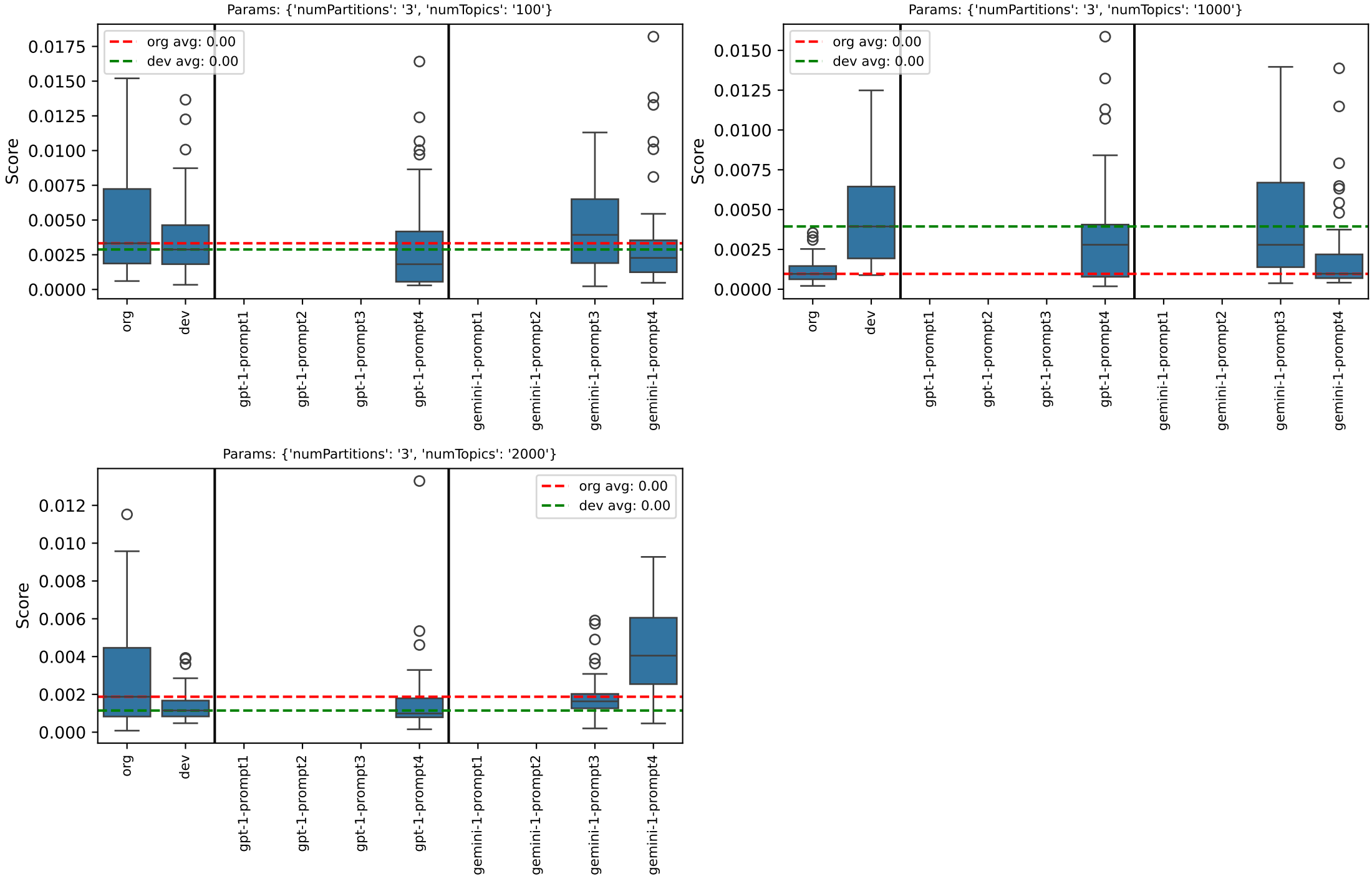
PerSupplierStr: 'CREATE', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '10', 'messageSize': '5000', 'stressLevel': 'DEV', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize': '10000'



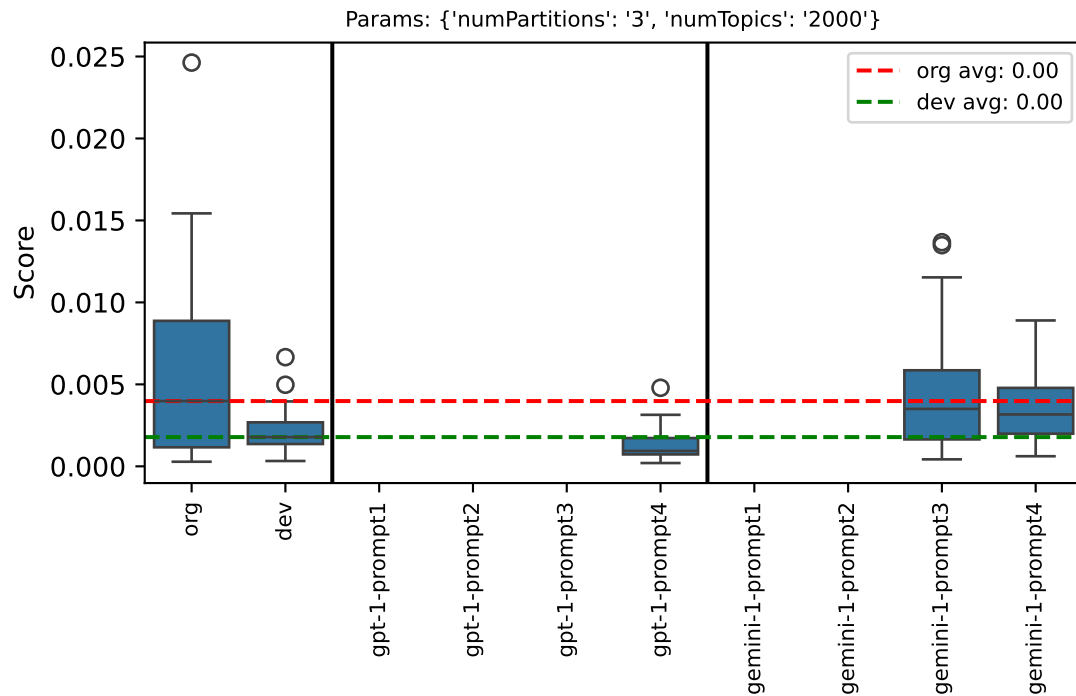
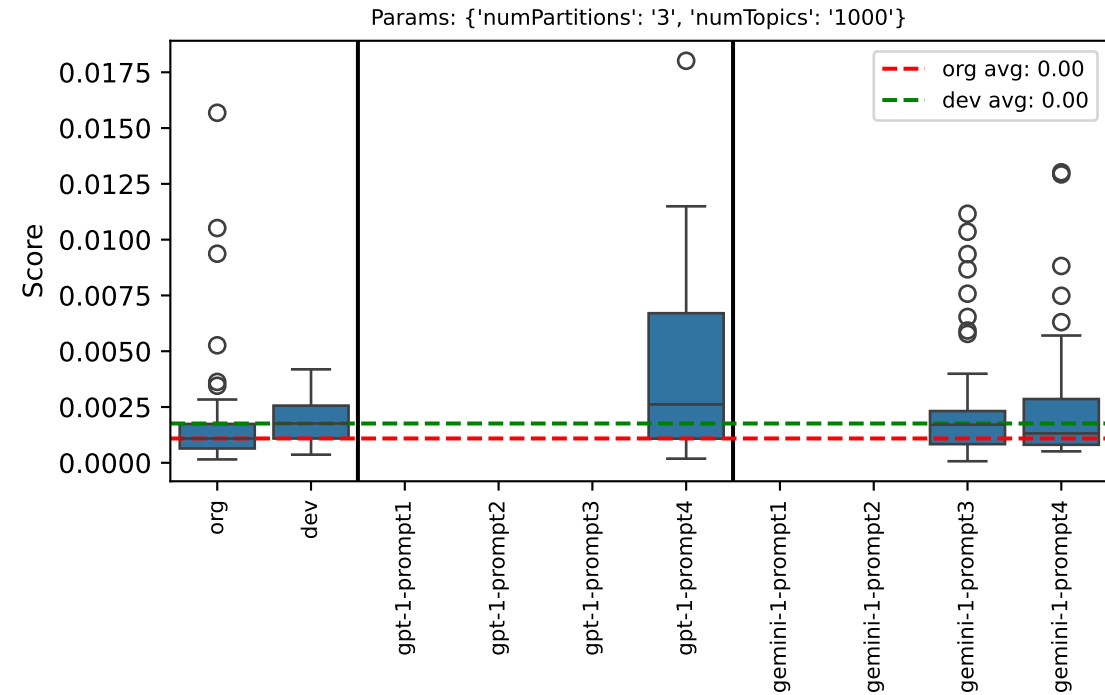
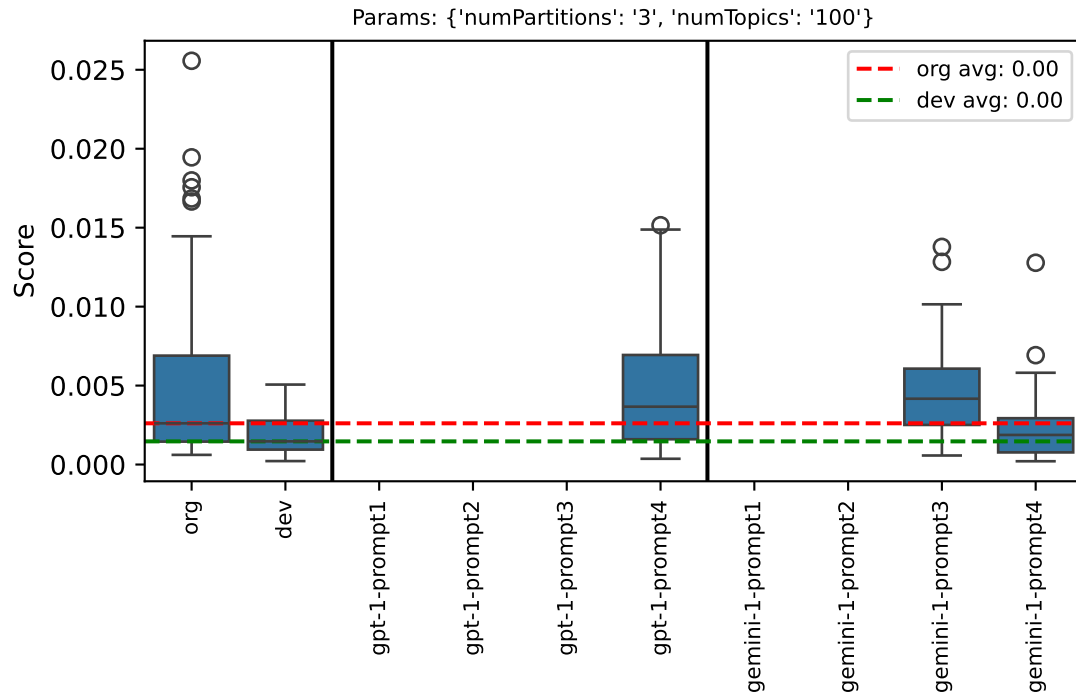
PerSupplierStr: 'CREATE', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize': '5000', 'stressLevel': 'DEV', 'bytes': 'ONES', 'compressionType': 'ZSTD', 'maxBatchSize': '50', 'messageSize': '10000'



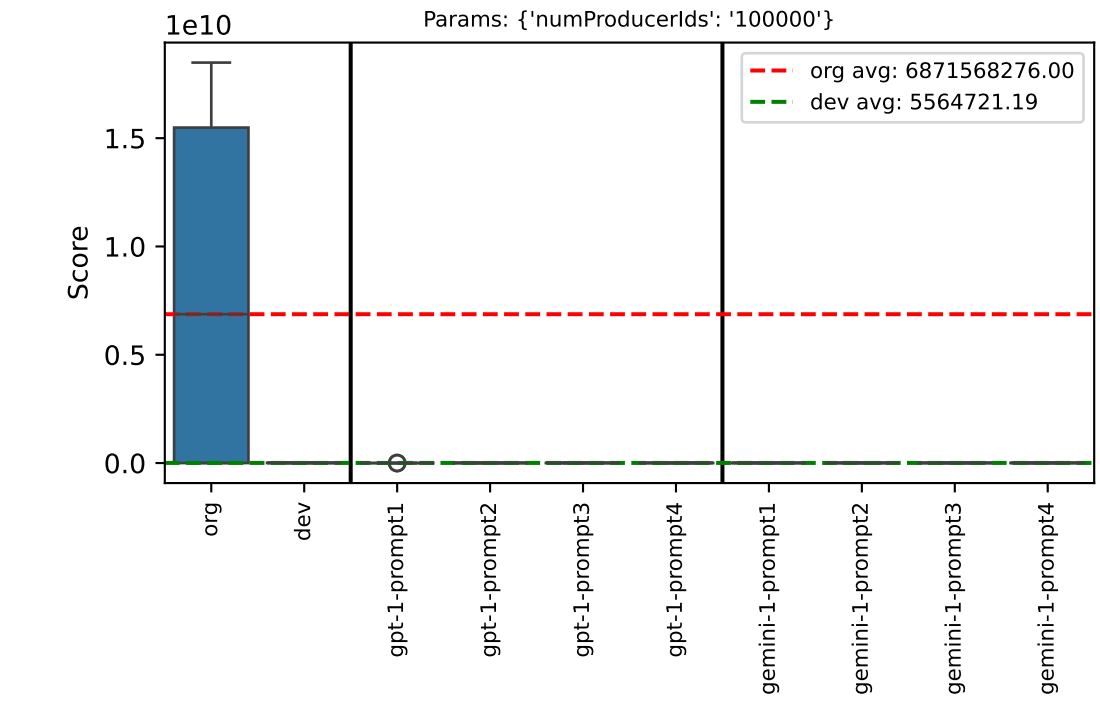
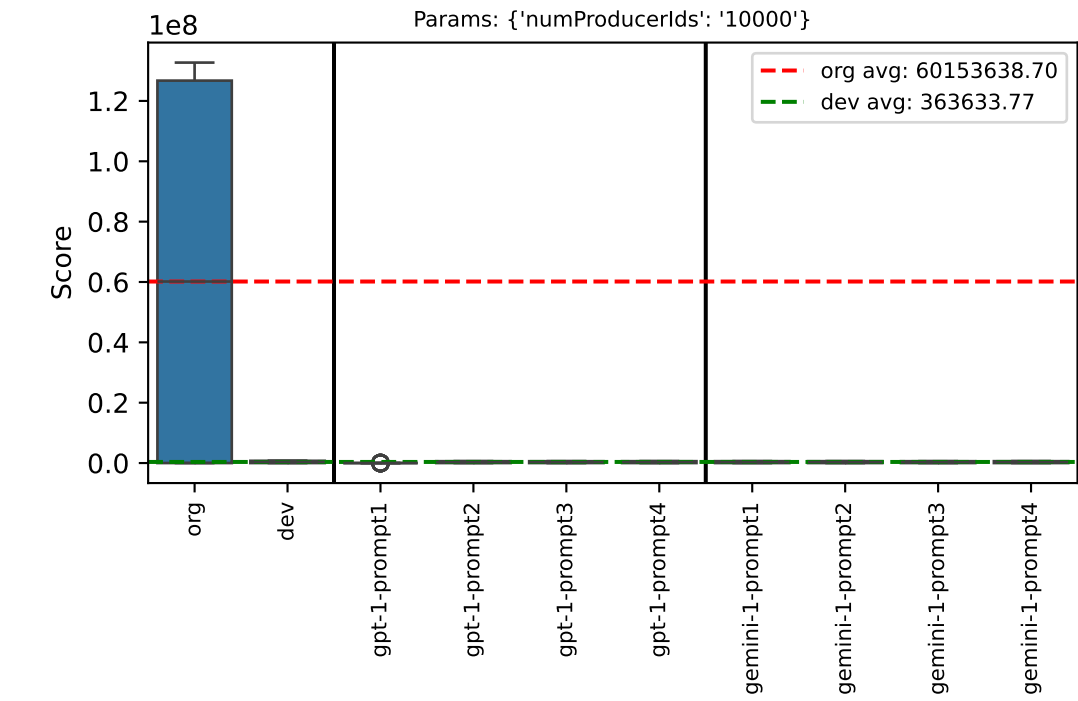
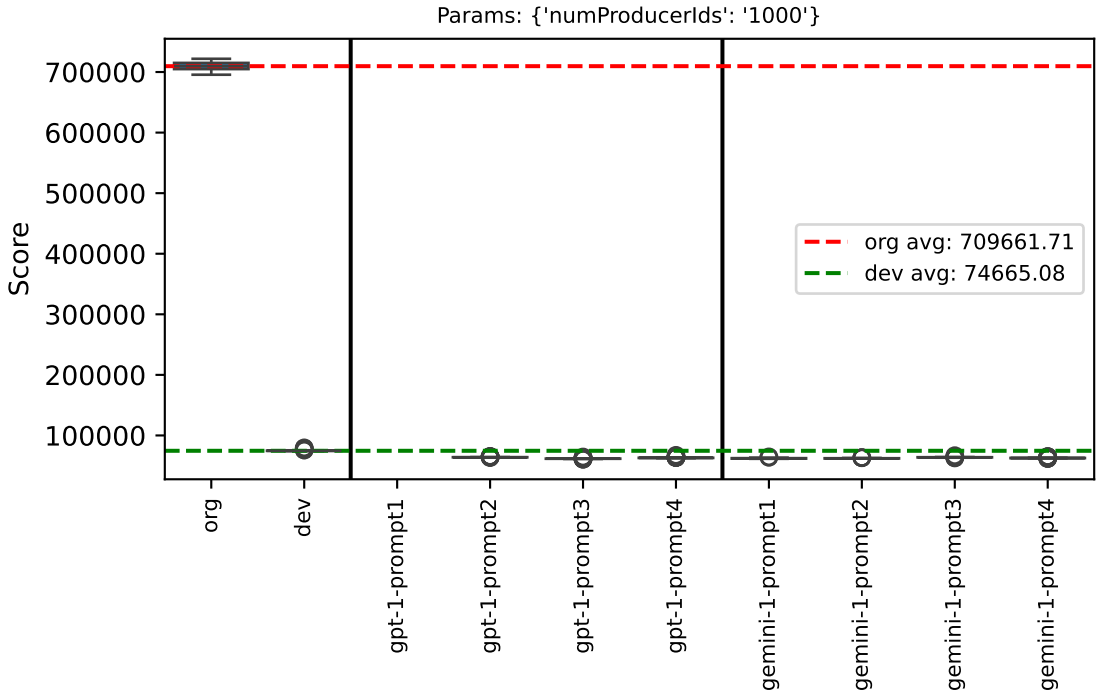
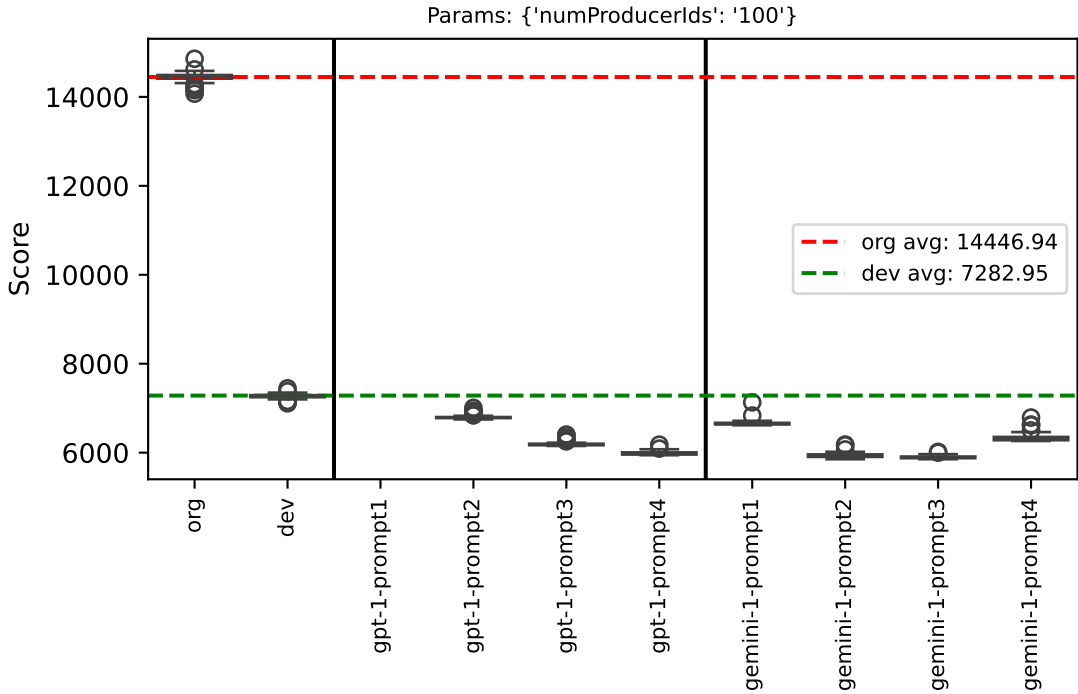
Benchmark: org.apache.kafka.jmh.server.CheckpointBench.measureCheckpointHighWatermarks



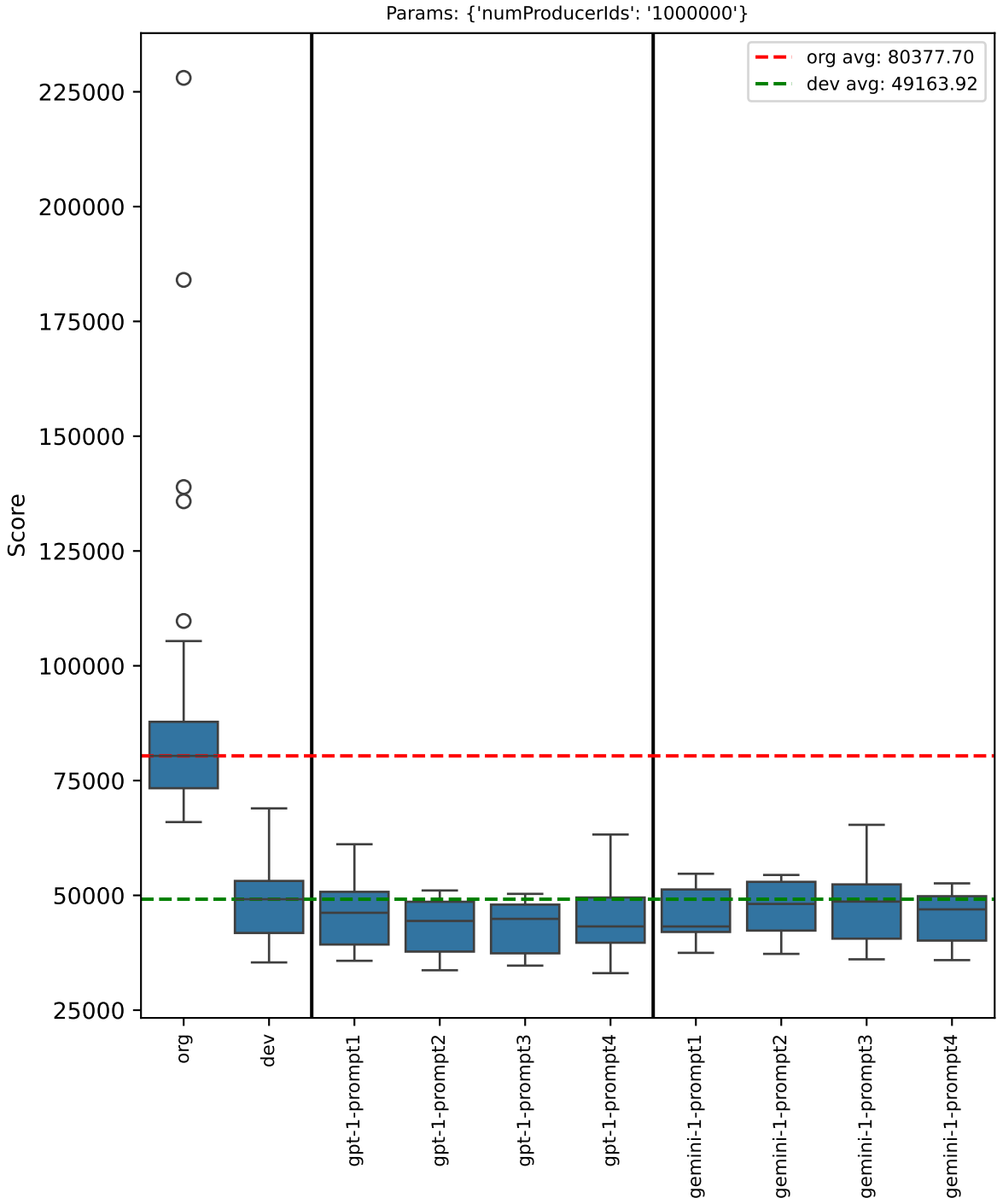
Benchmark: org.apache.kafka.jmh.server.CheckpointBench.measureCheckpointLogStartOffsets



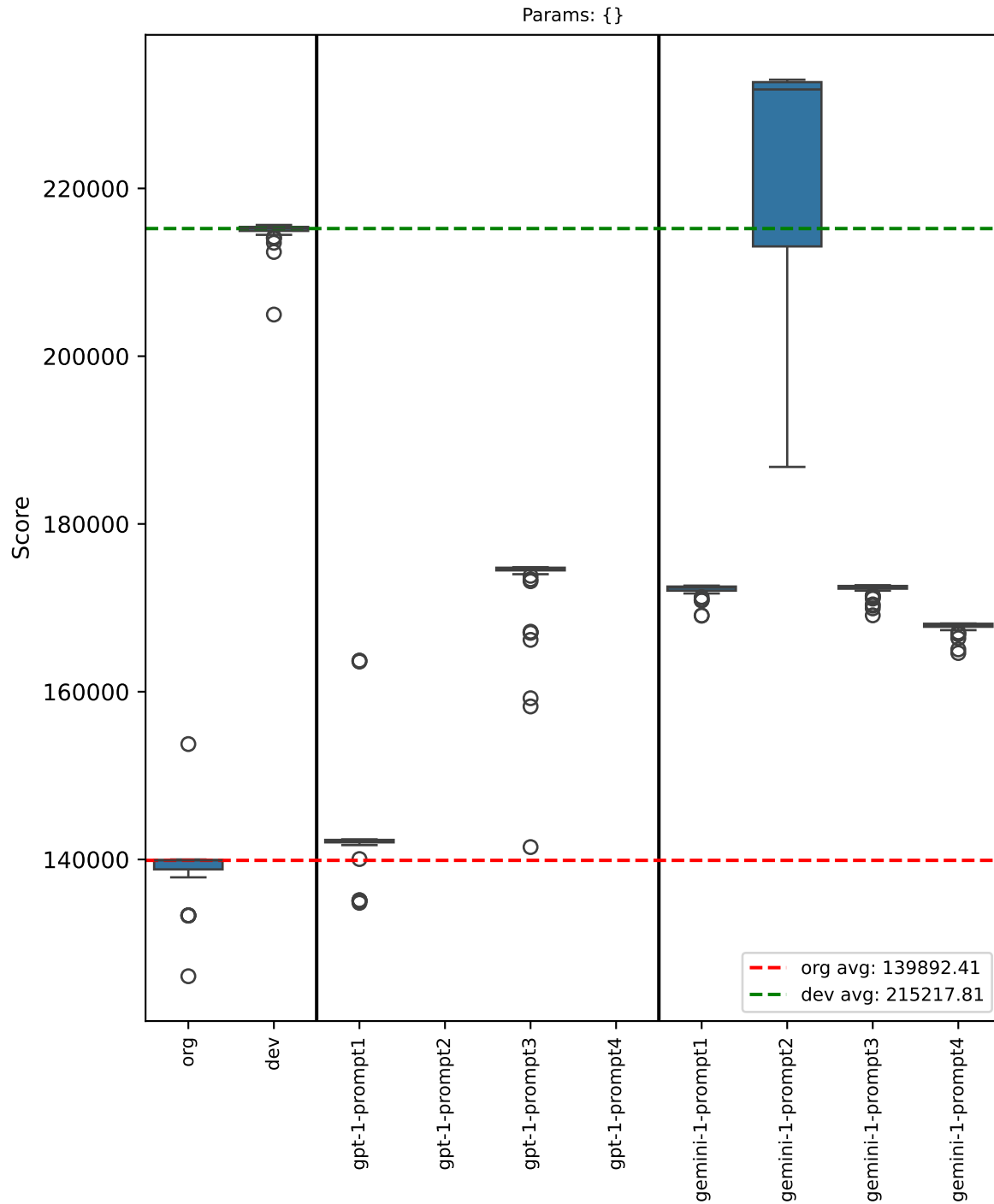
Benchmark: org.apache.kafka.jmh.storage.ProducerStateManagerBench.testDeleteExpiringIds



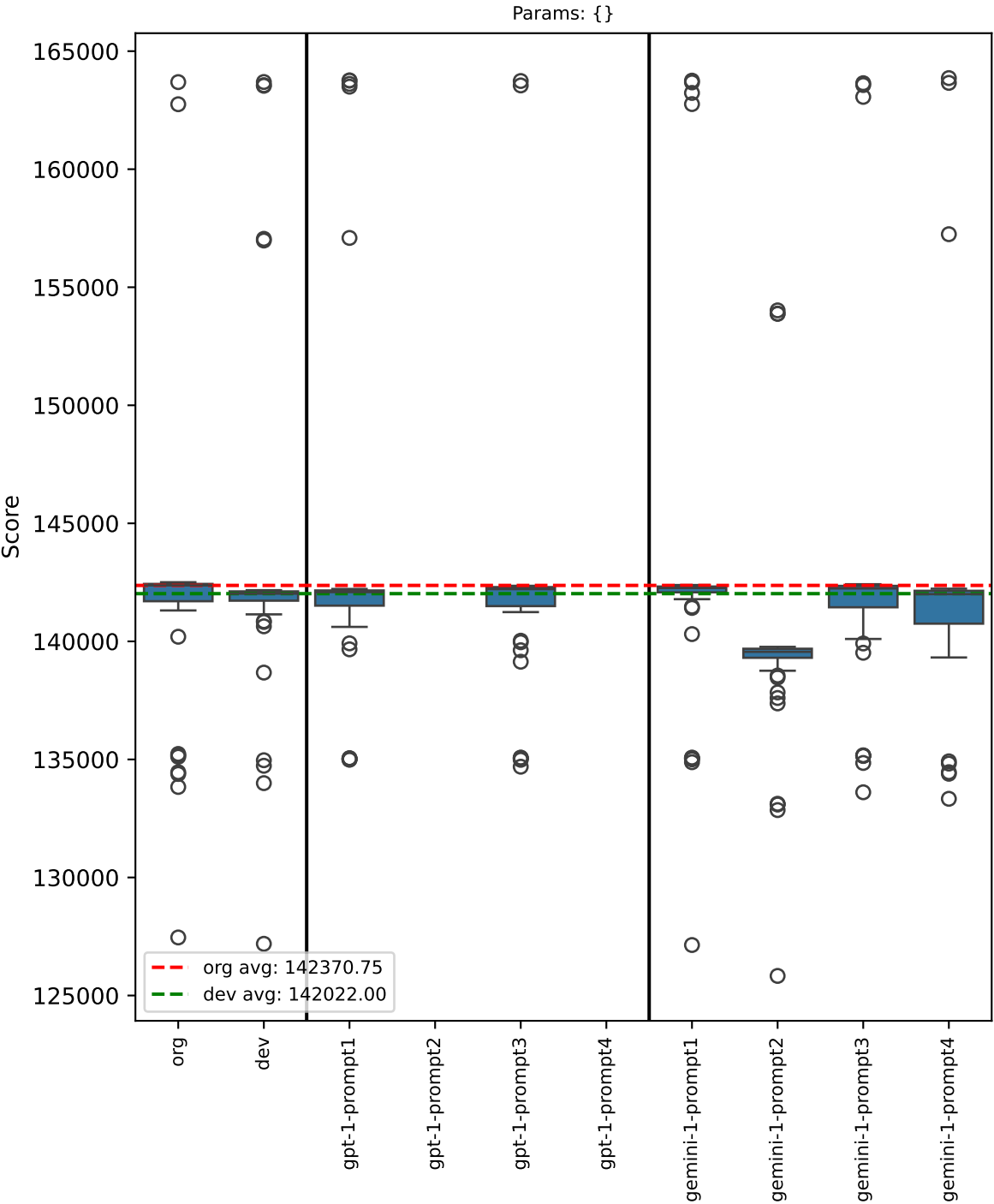
Benchmark: org.apache.kafka.jmh.storage.ProducerStateManagerBench.testDeleteExpiringIds



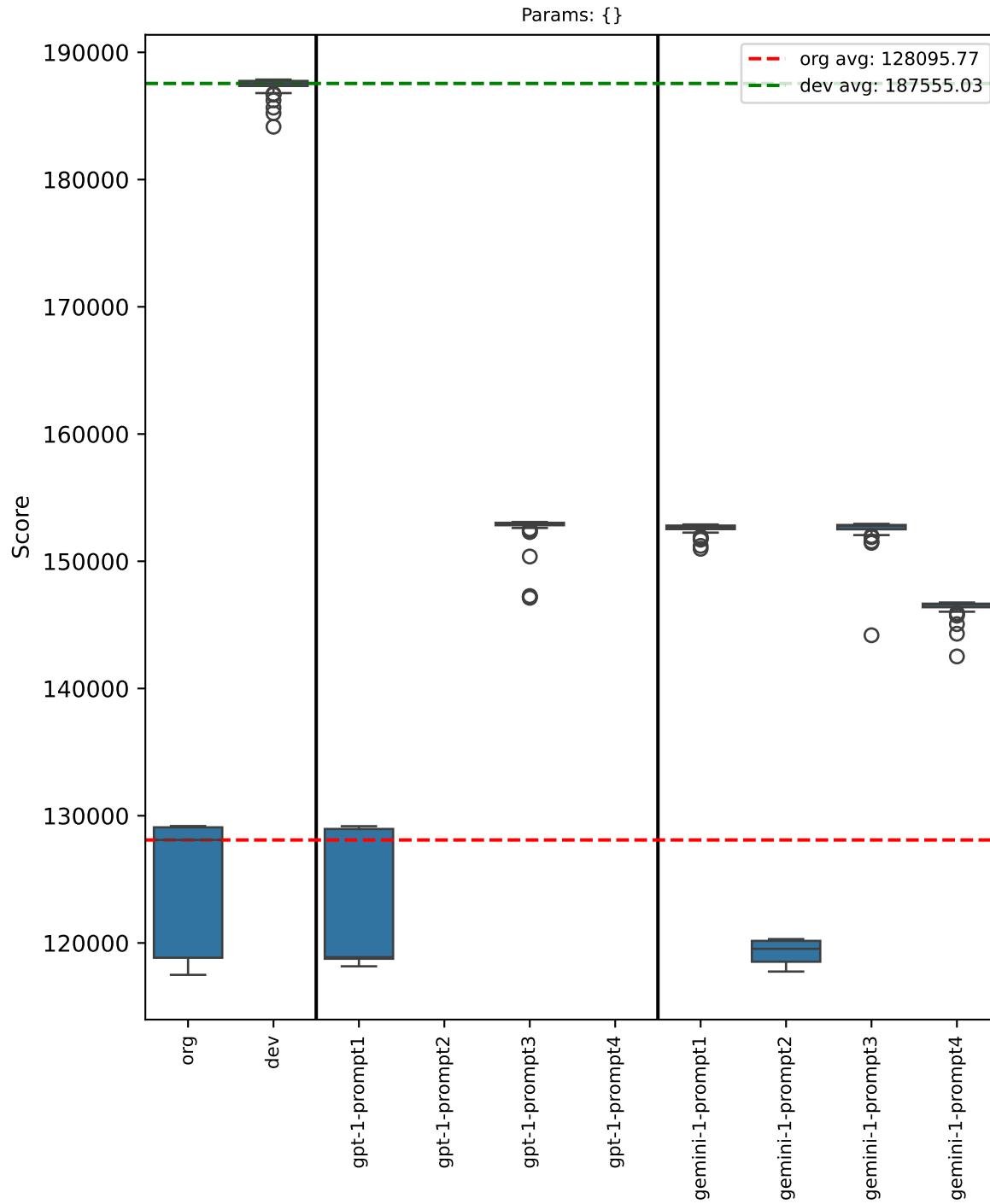
Benchmark: org.apache.kafka.jmh.util.ByteUtilsBenchmark.testSizeOfUnsignedVarint



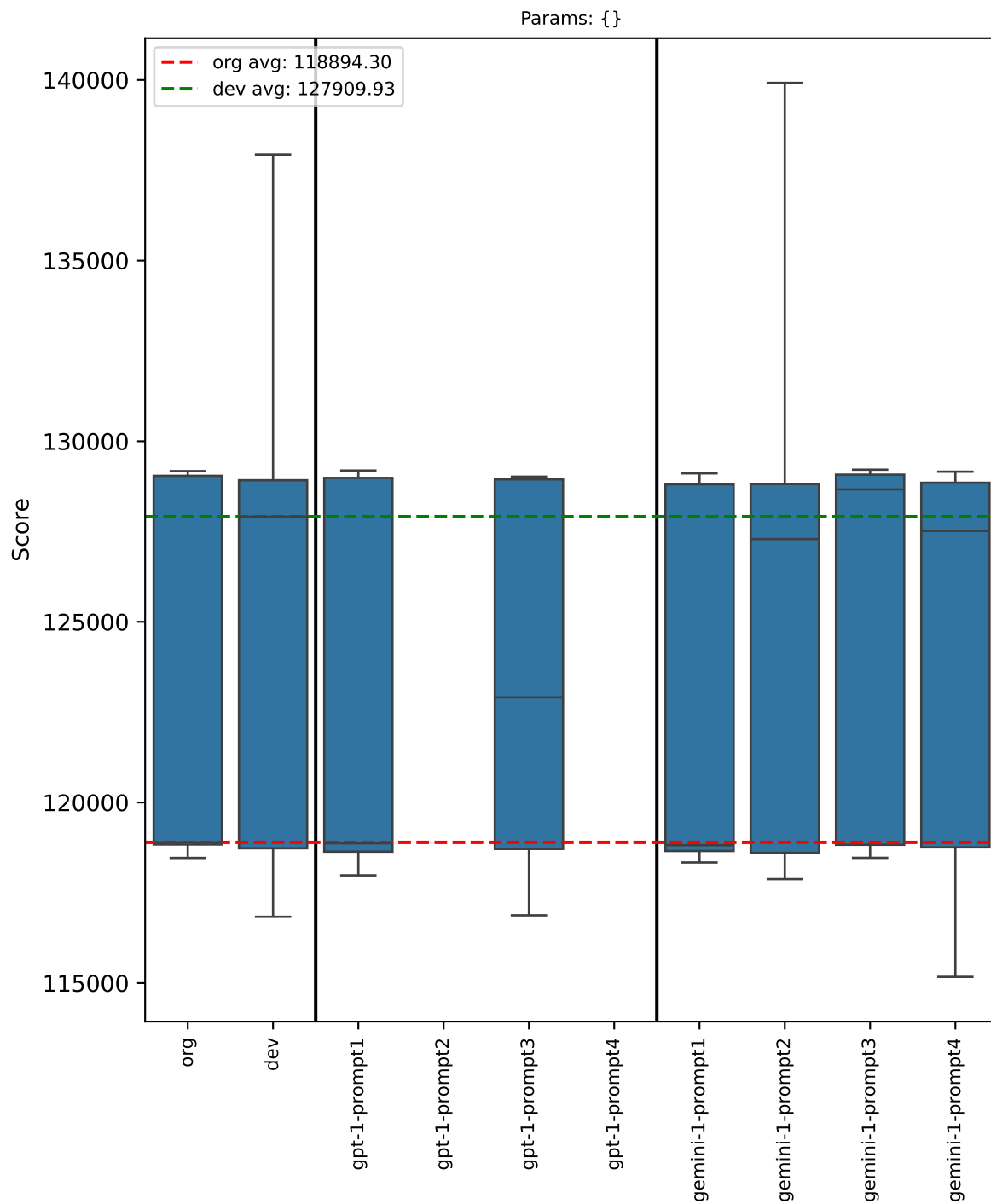
Benchmark: org.apache.kafka.jmh.util.ByteUtilsBenchmark.testSizeOfUnsignedVarintSimple



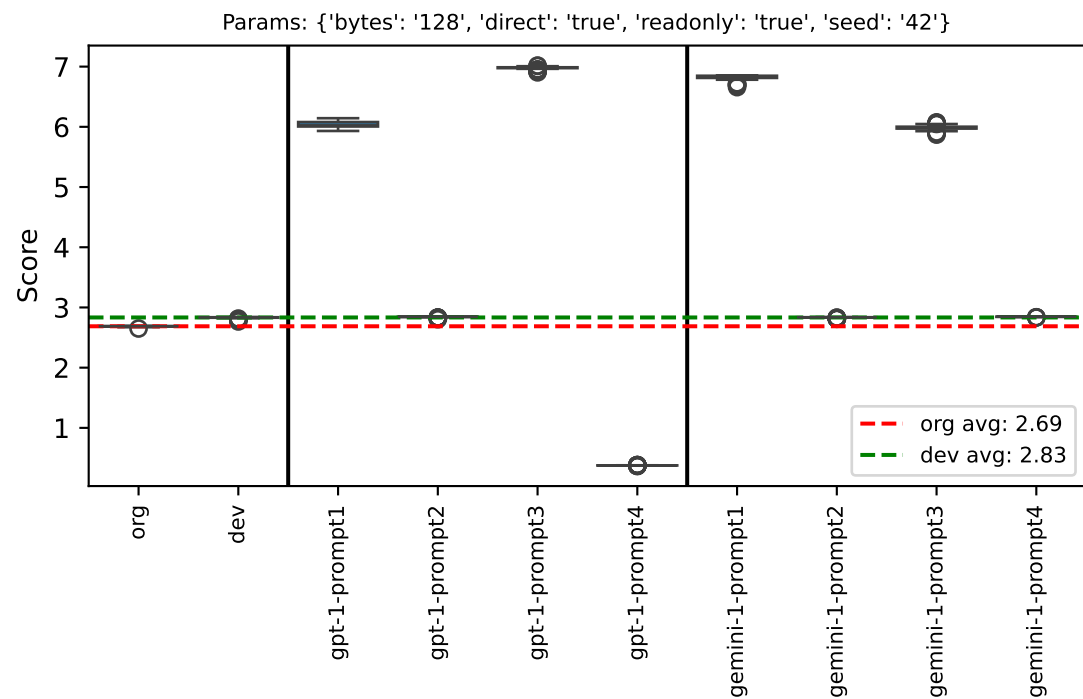
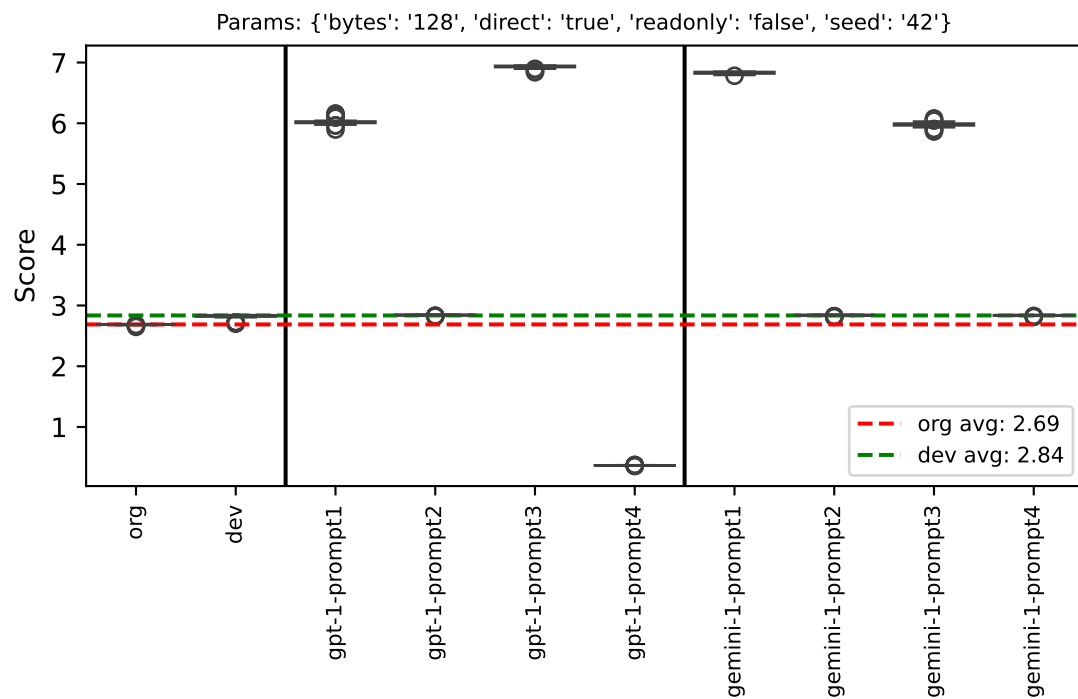
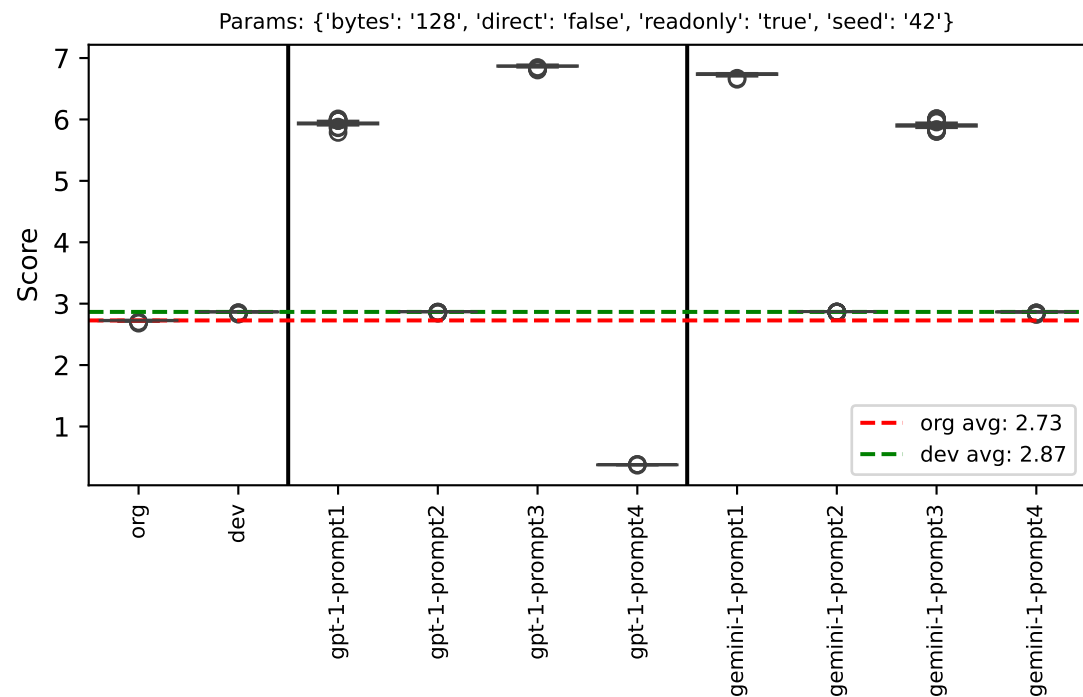
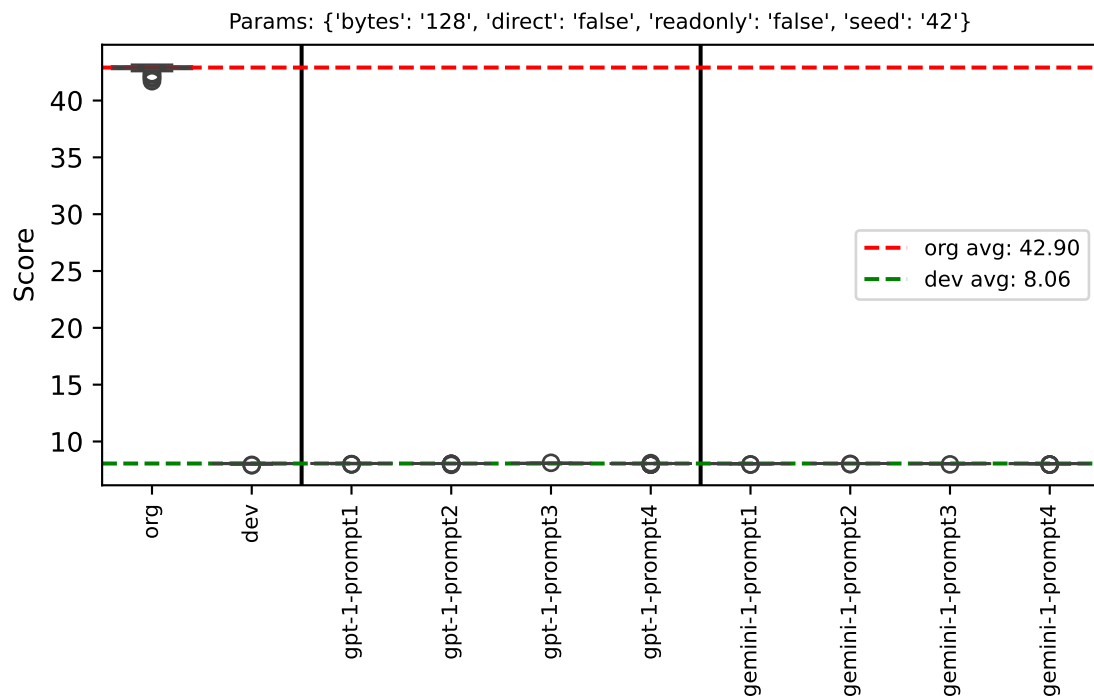
Benchmark: org.apache.kafka.jmh.util.ByteUtilsBenchmark.testSizeOfVarlong



Benchmark: org.apache.kafka.jmh.util.ByteUtilsBenchmark.testSizeOfVarlongSimple

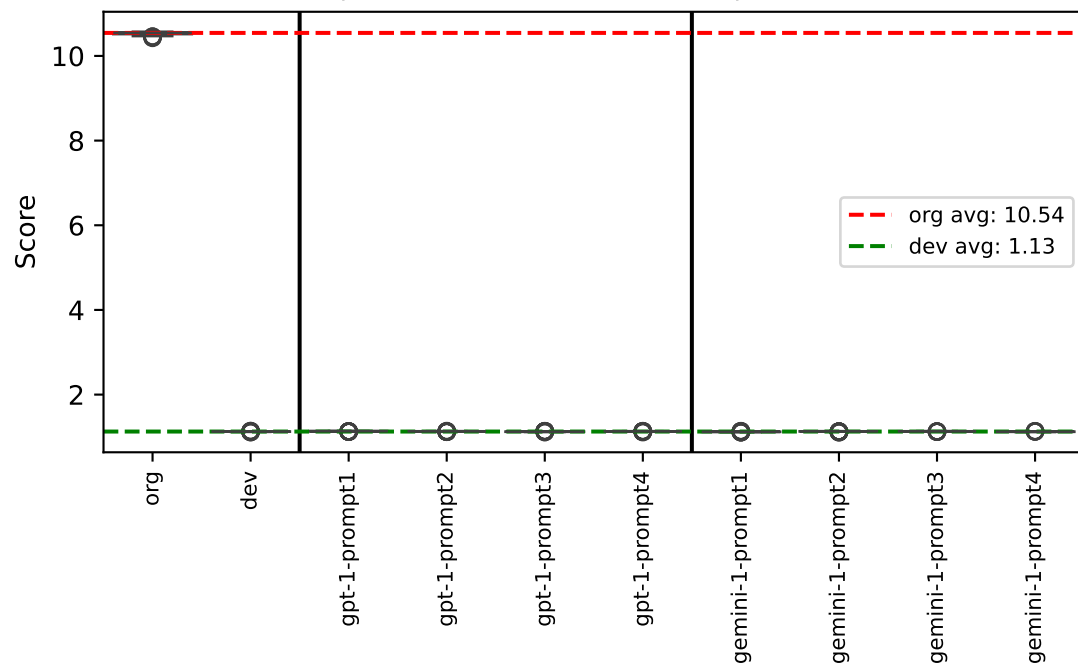


Benchmark: org.apache.kafka.jmh.util.Crc32CBenchmark.checksum

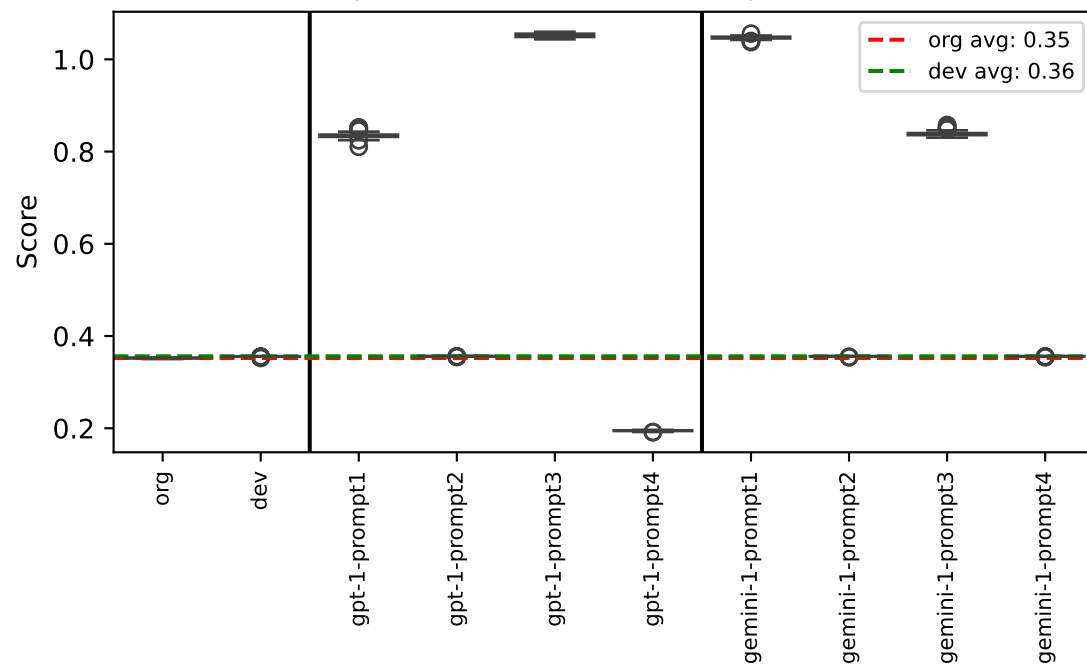


Benchmark: org.apache.kafka.jmh.util.Crc32CBenchmark.checksum

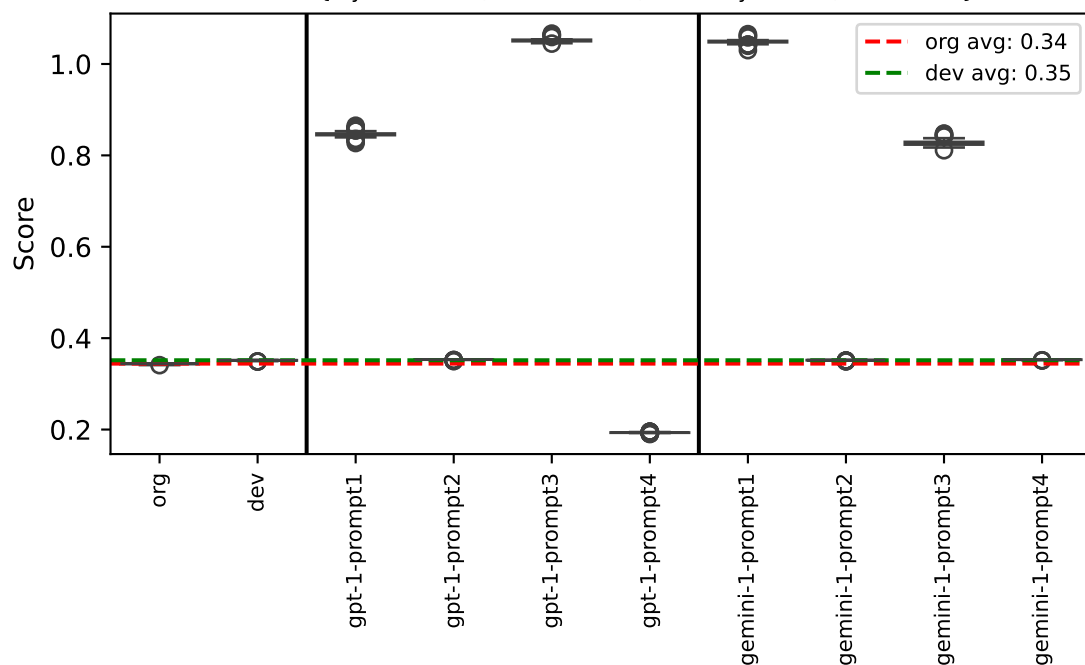
Params: {'bytes': '1024', 'direct': 'false', 'readonly': 'false', 'seed': '42'}



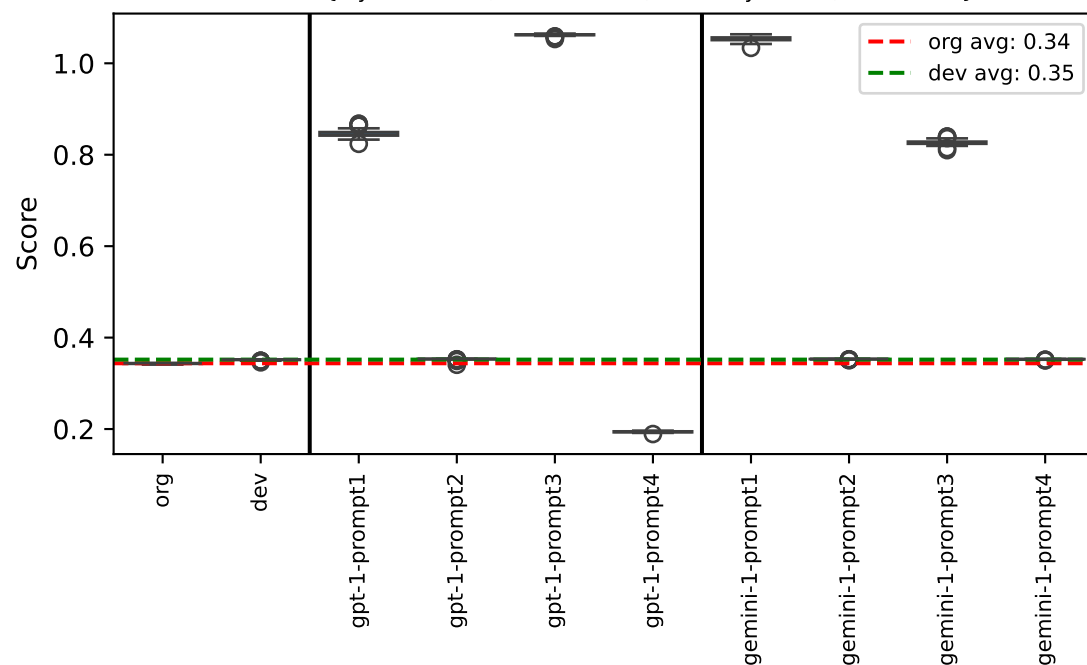
Params: {'bytes': '1024', 'direct': 'false', 'readonly': 'true', 'seed': '42'}



Params: {'bytes': '1024', 'direct': 'true', 'readonly': 'false', 'seed': '42'}

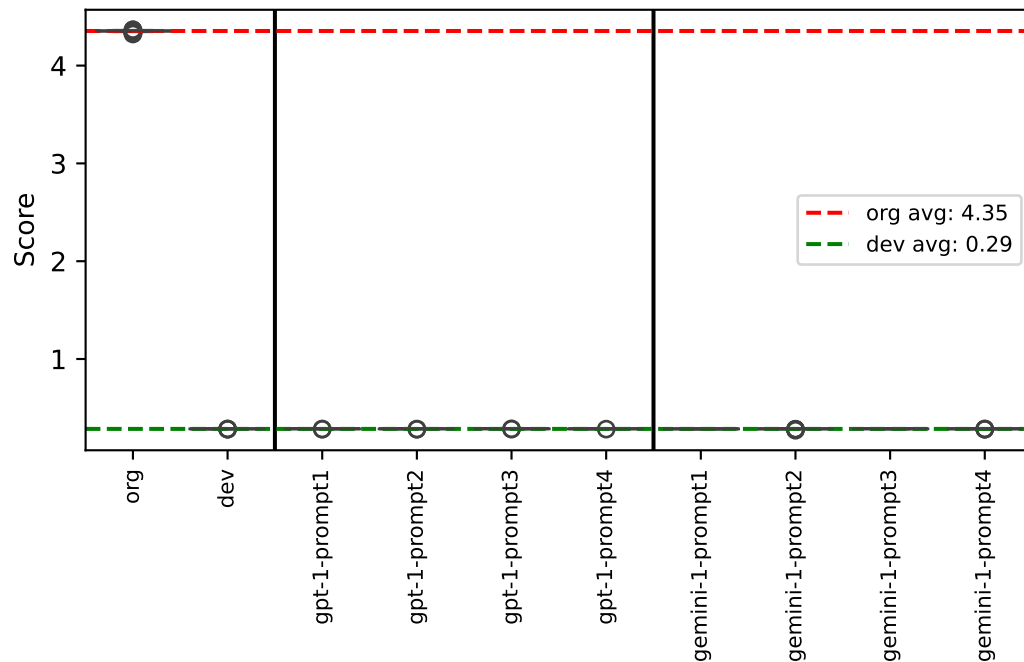


Params: {'bytes': '1024', 'direct': 'true', 'readonly': 'true', 'seed': '42'}

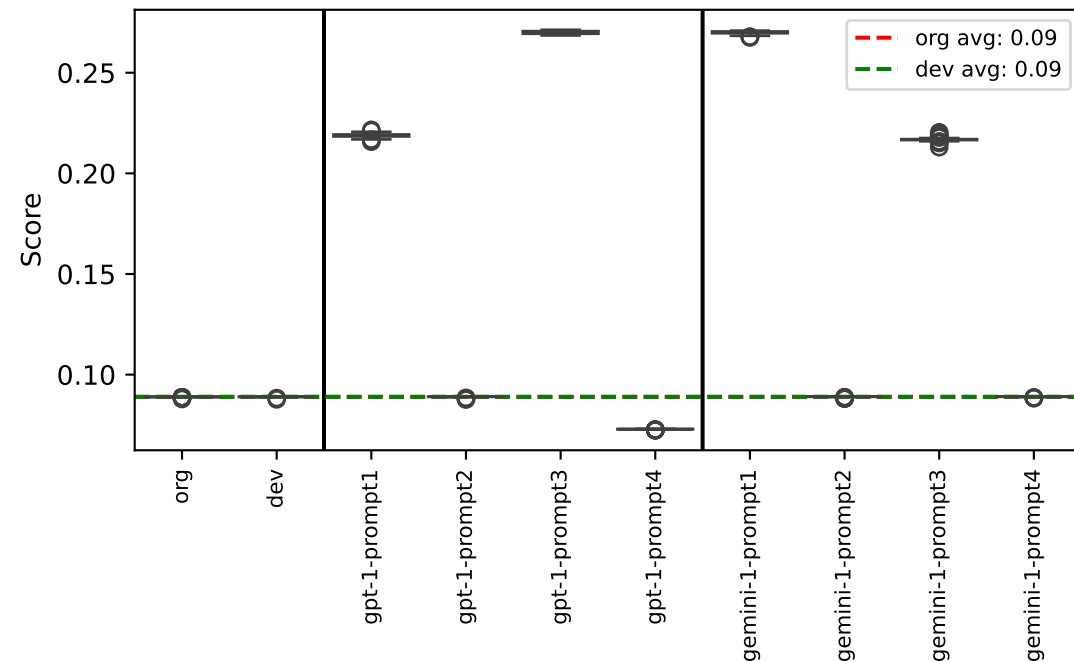


Benchmark: org.apache.kafka.jmh.util.Crc32CBenchmark.checksum

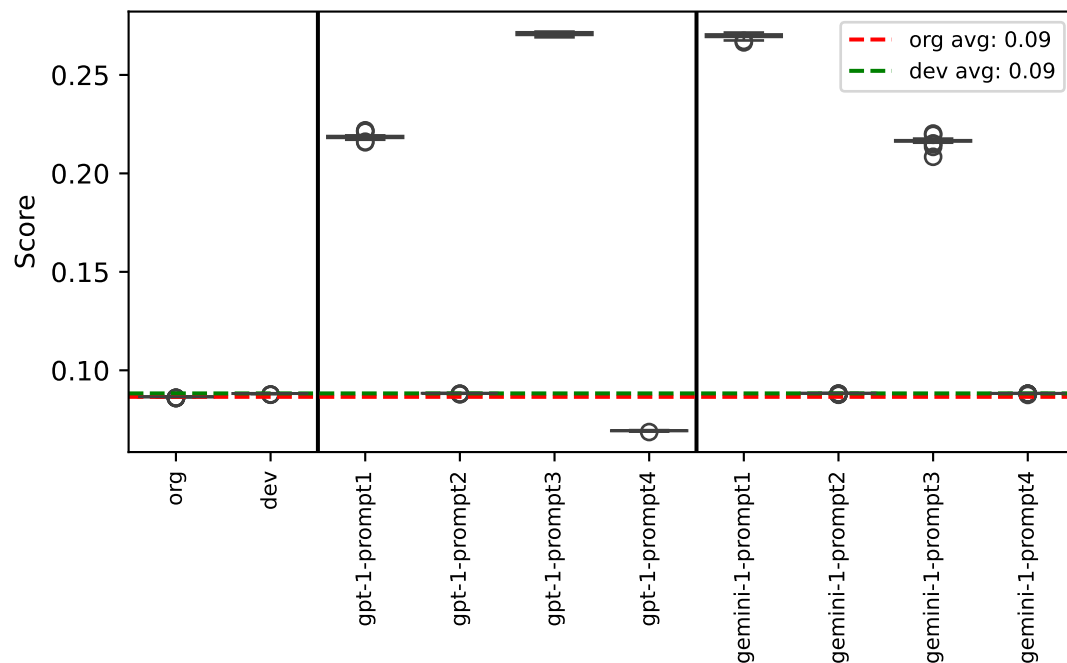
Params: {'bytes': '4096', 'direct': 'false', 'readonly': 'false', 'seed': '42'}



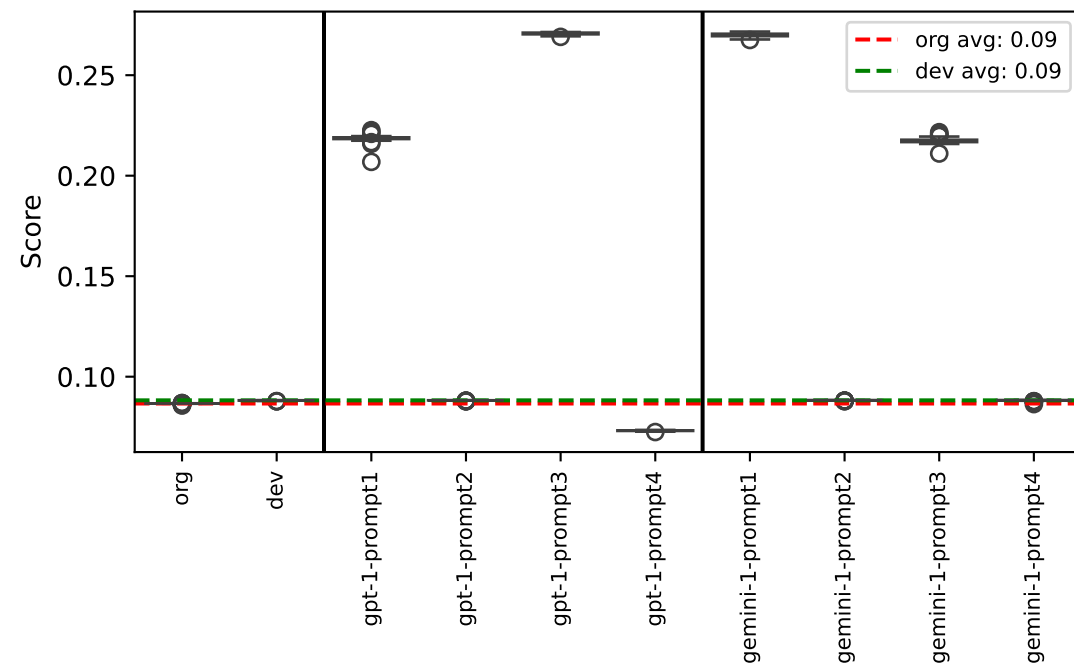
Params: {'bytes': '4096', 'direct': 'false', 'readonly': 'true', 'seed': '42'}



Params: {'bytes': '4096', 'direct': 'true', 'readonly': 'false', 'seed': '42'}

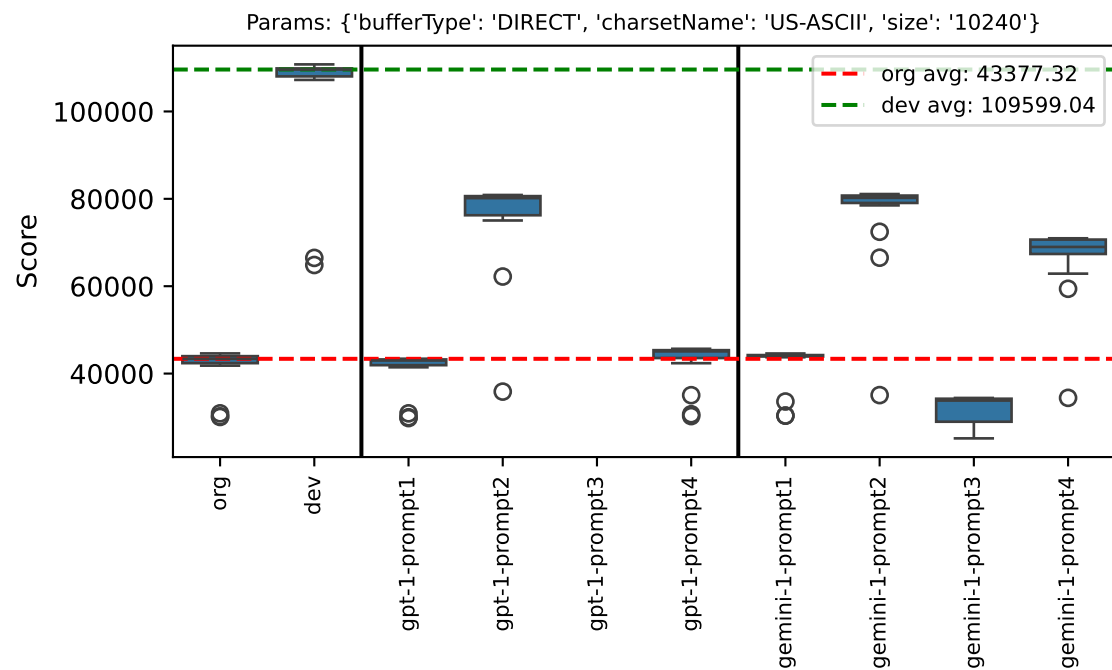
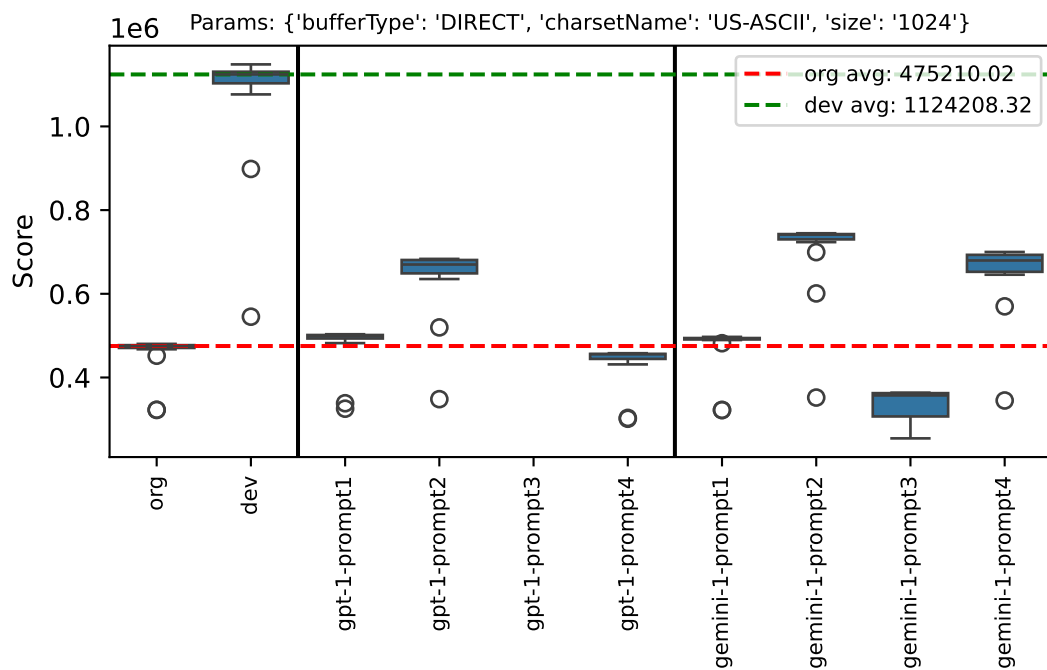
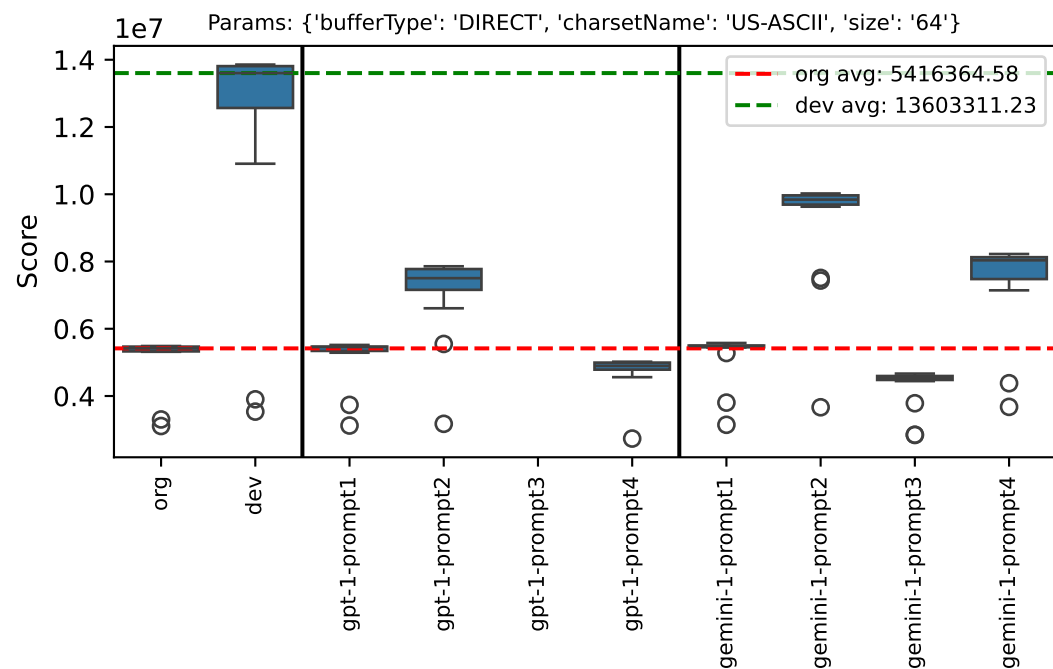
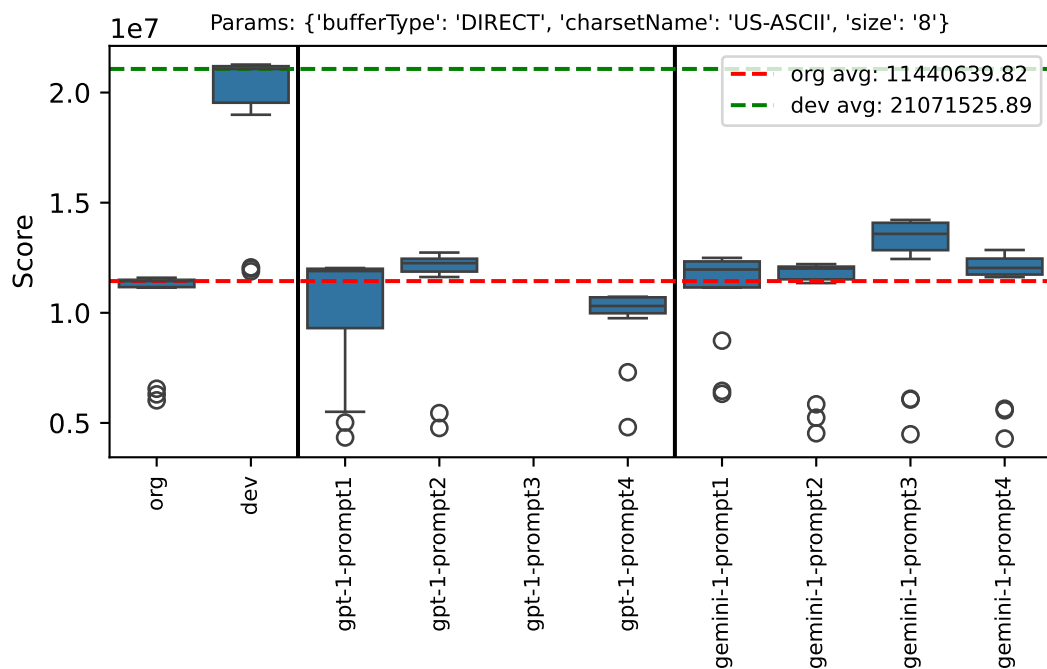


Params: {'bytes': '4096', 'direct': 'true', 'readonly': 'true', 'seed': '42'}

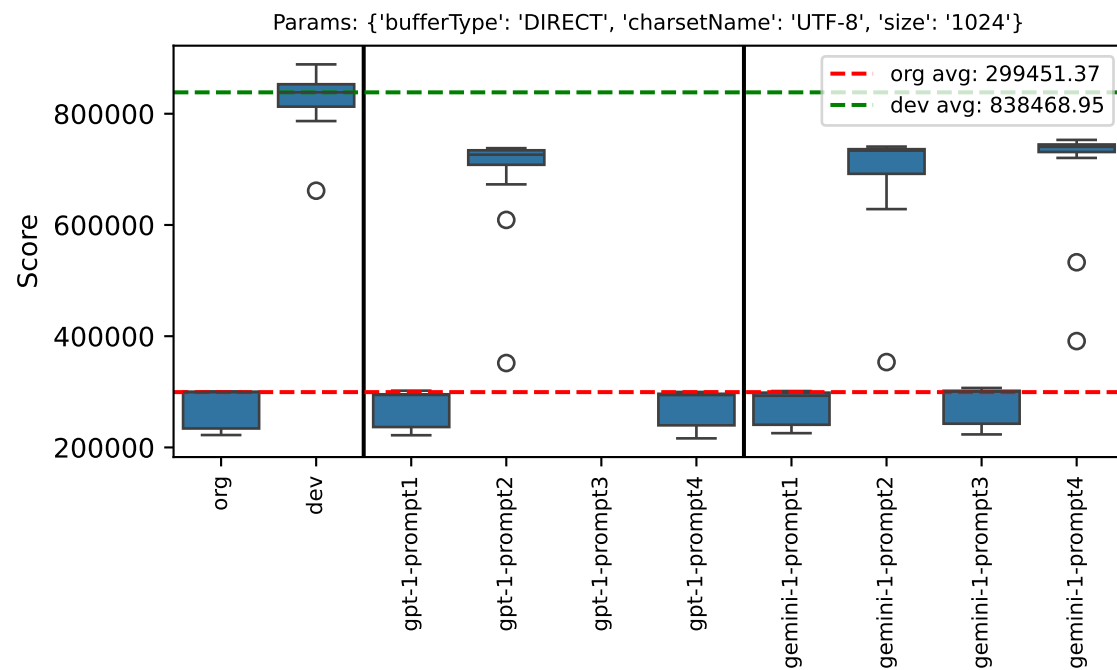
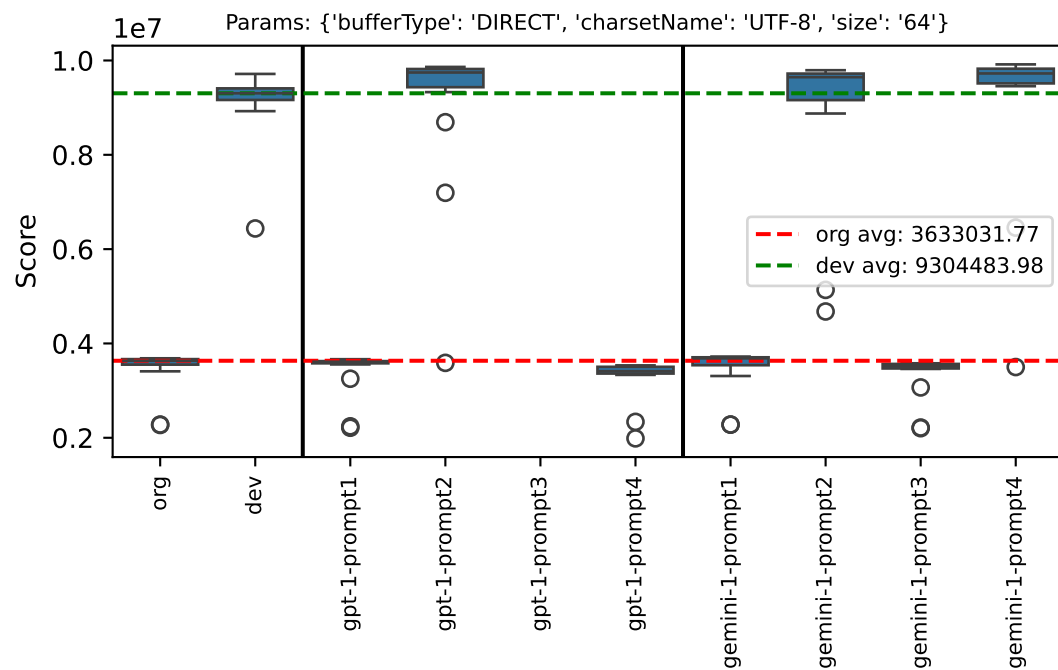
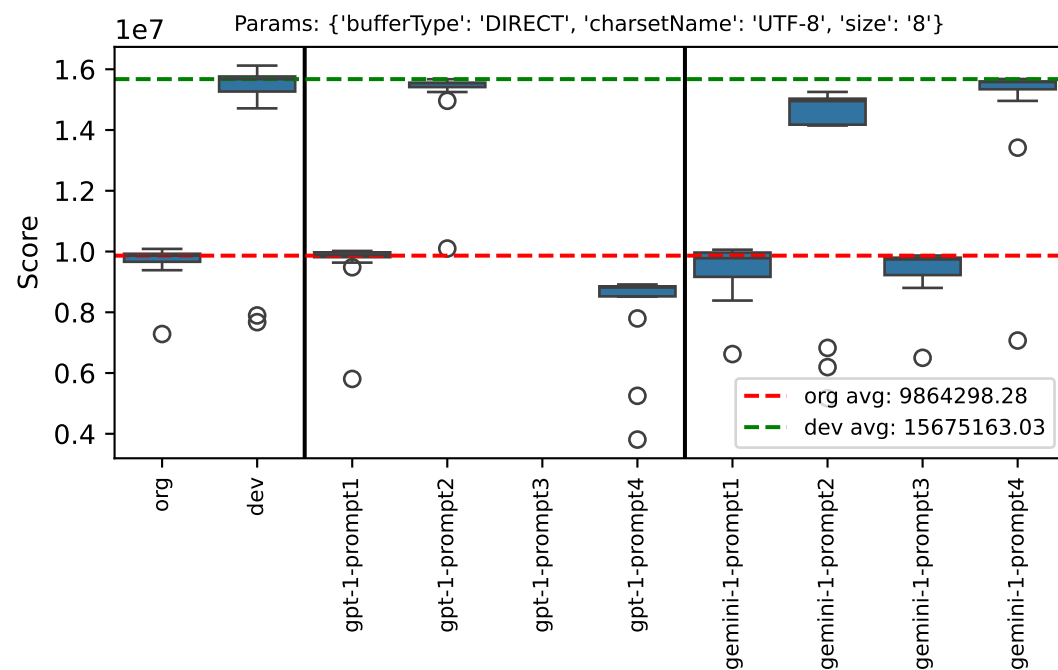
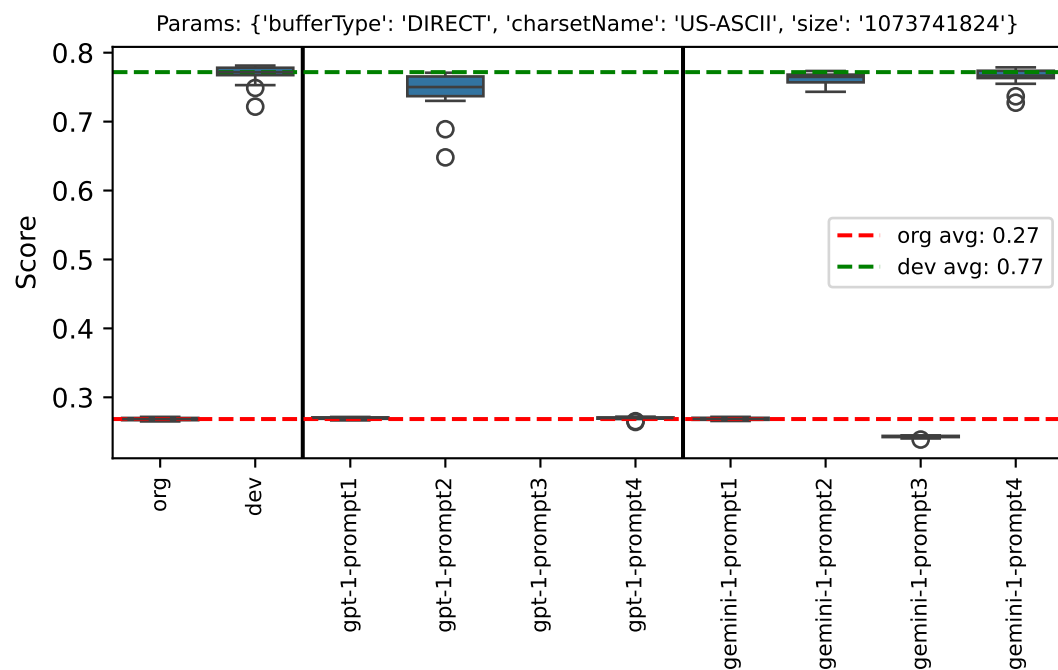


Project: netty

Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

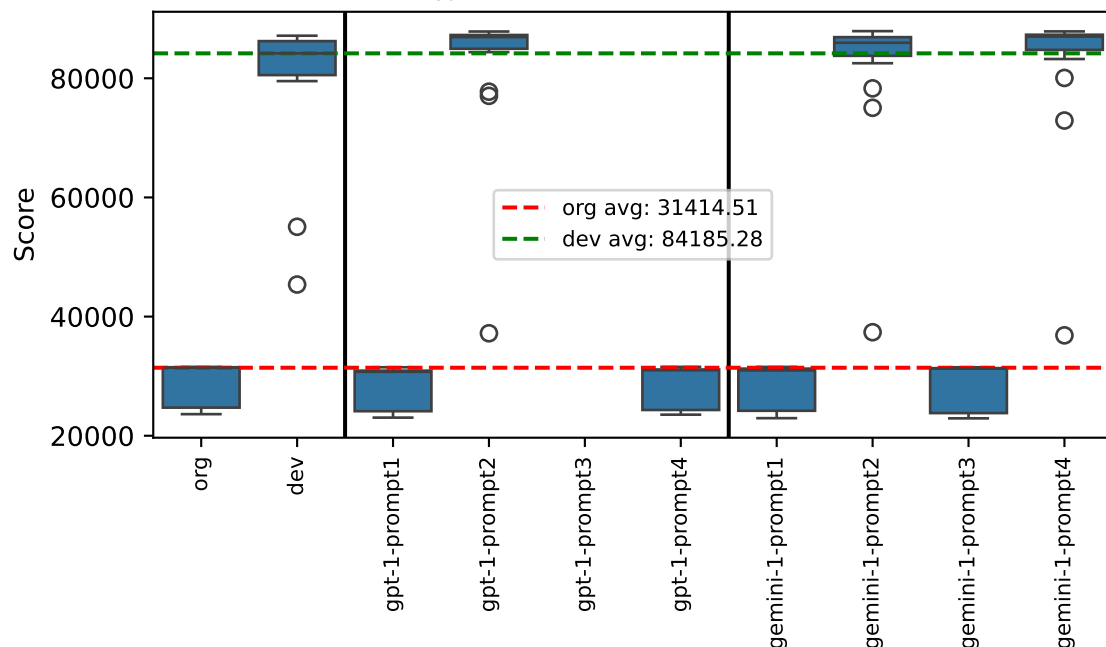


Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

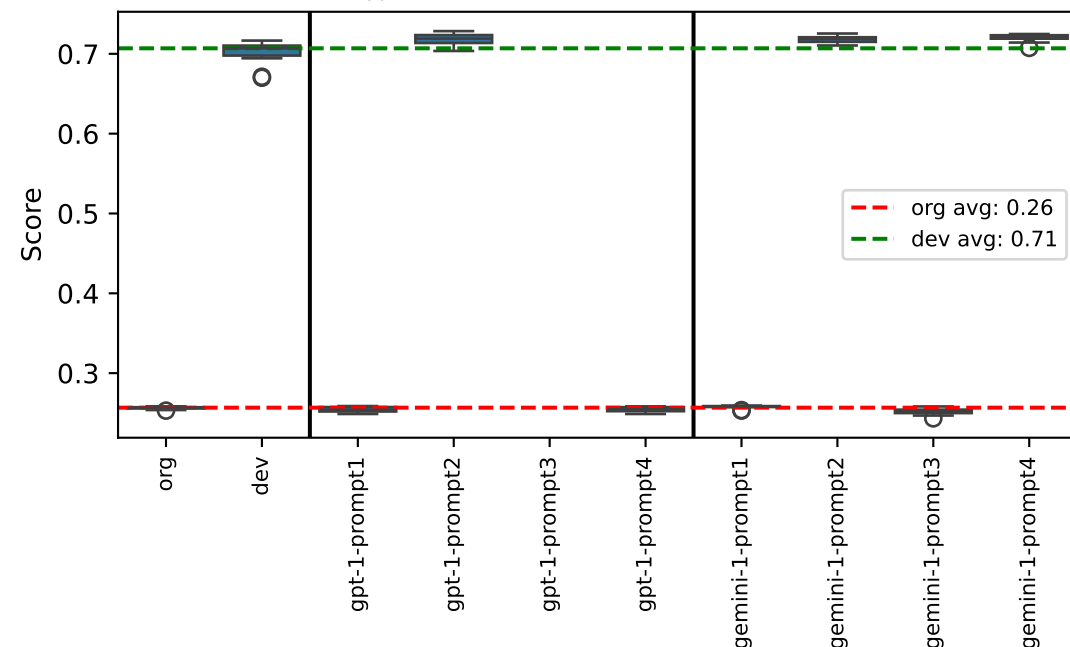


Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

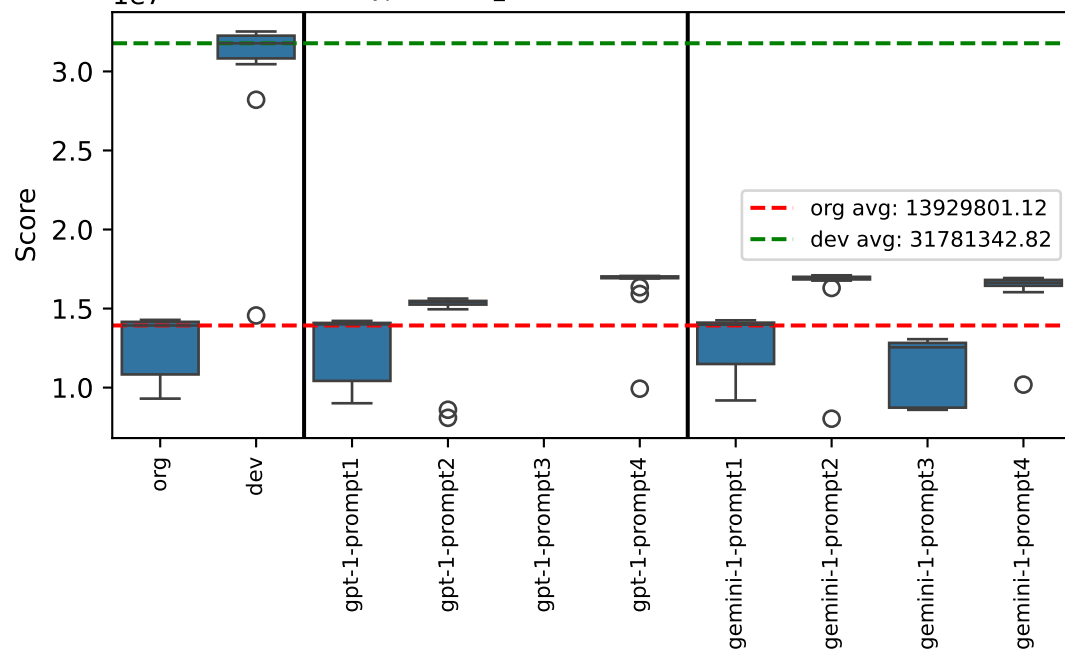
Params: {'bufferType': 'DIRECT', 'charsetName': 'UTF-8', 'size': '10240'}



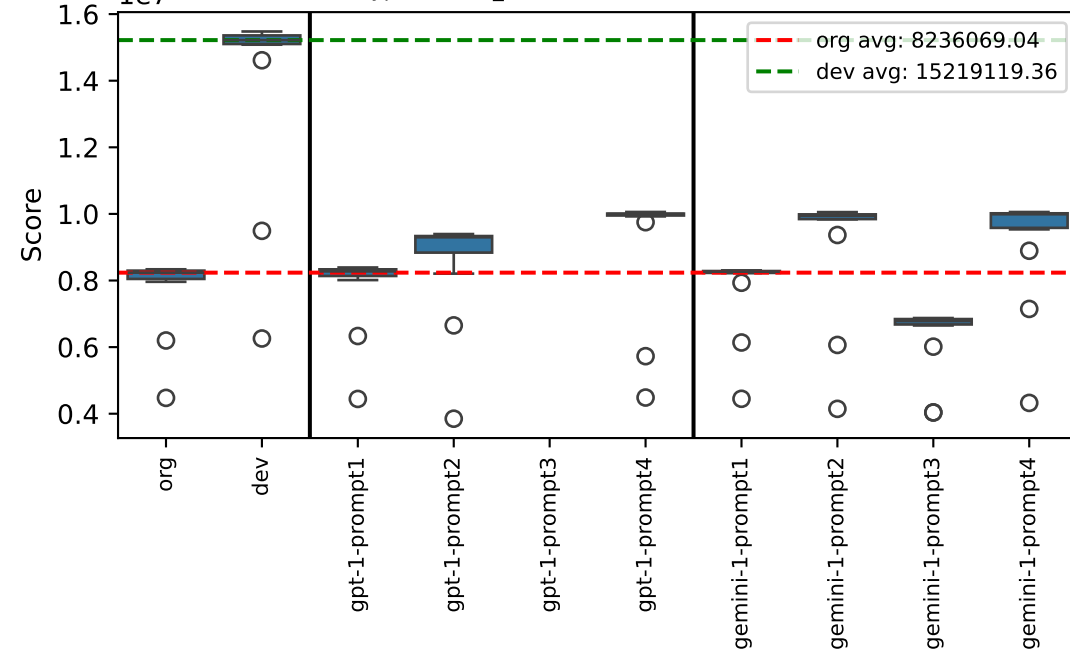
Params: {'bufferType': 'DIRECT', 'charsetName': 'UTF-8', 'size': '1073741824'}



1e7 Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'US-ASCII', 'size': '8'}

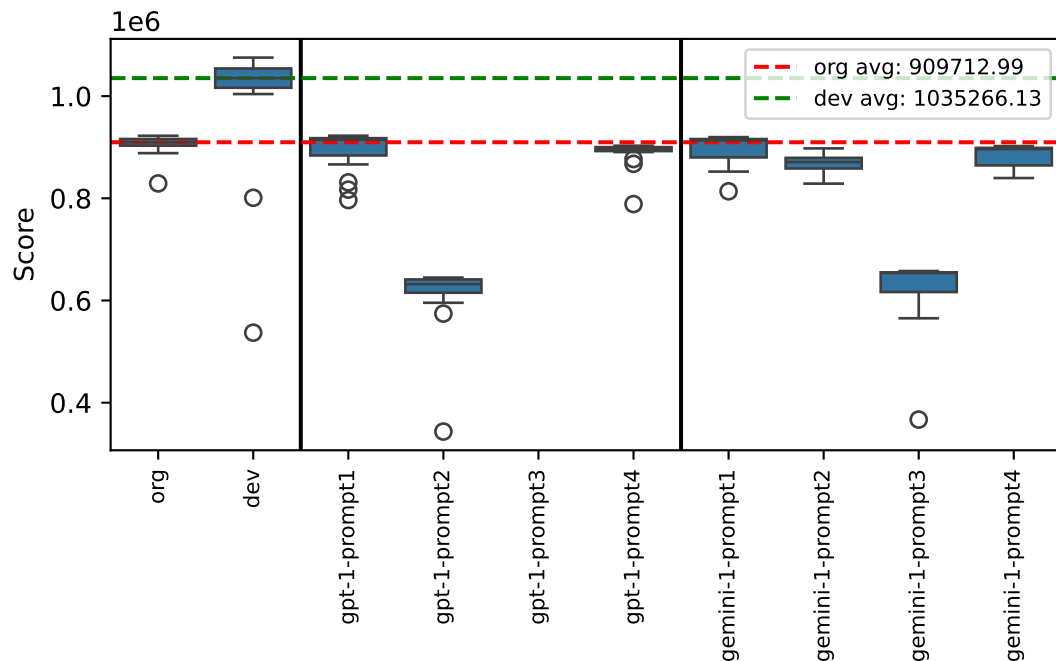


1e7 Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'US-ASCII', 'size': '64'}

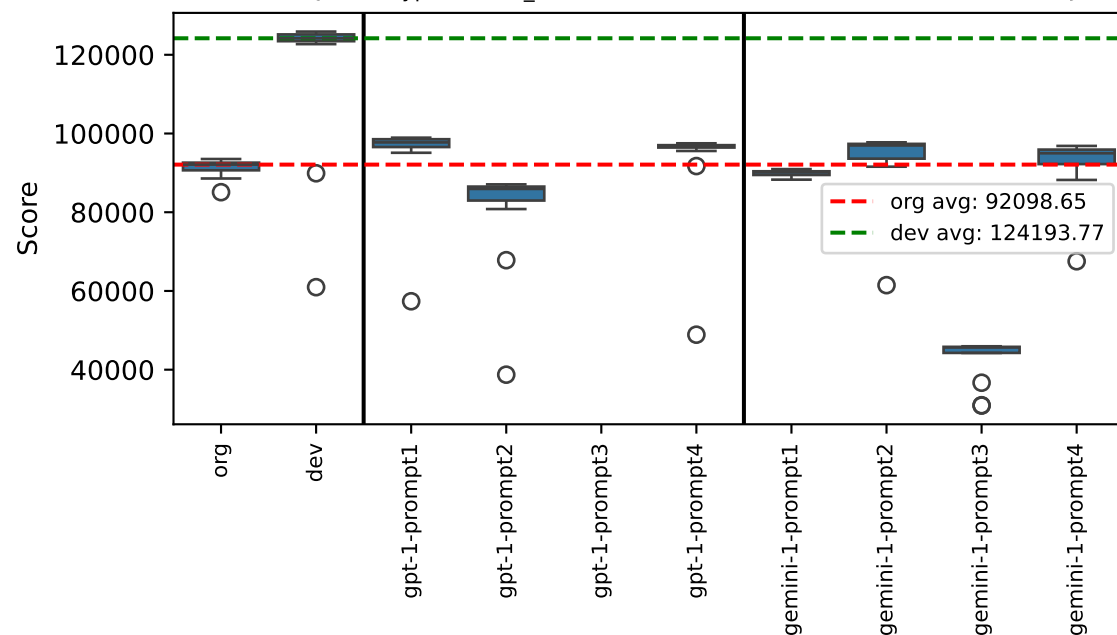


Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

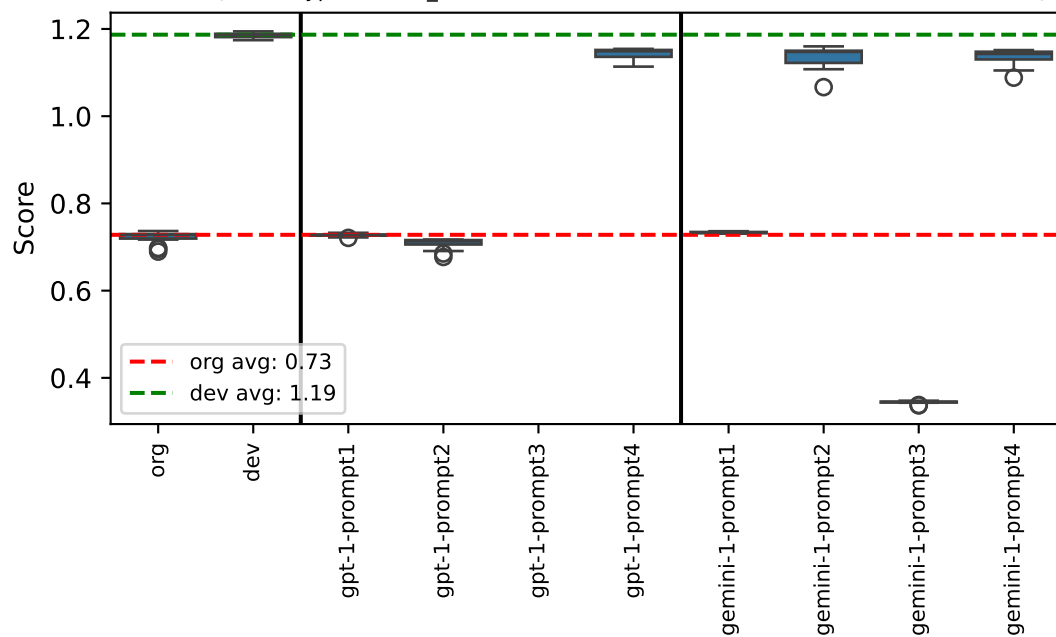
Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'US-ASCII', 'size': '1024'}



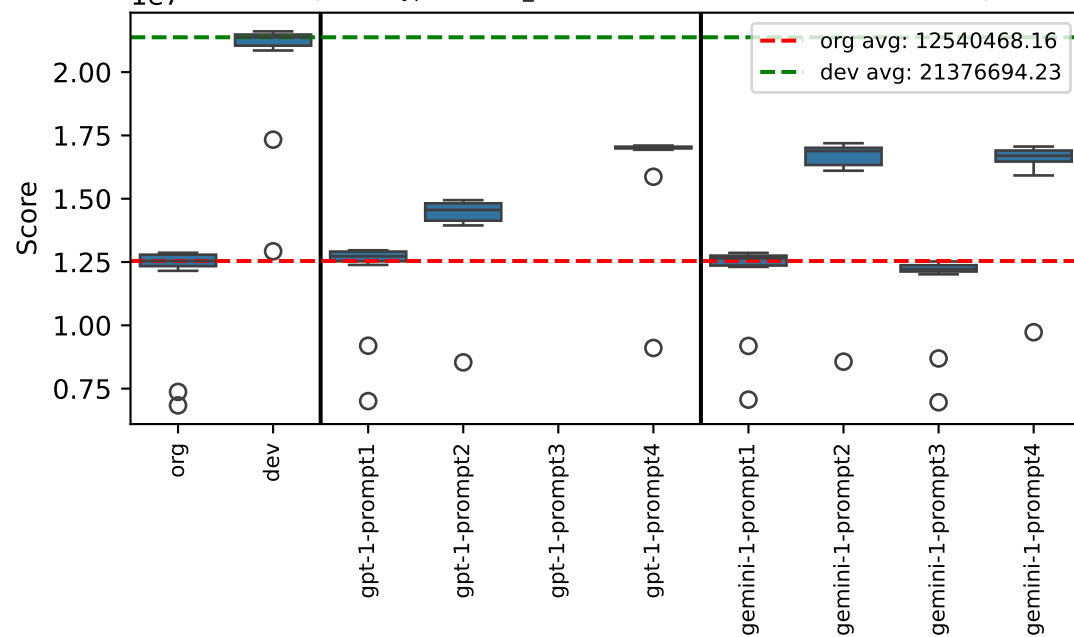
Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'US-ASCII', 'size': '10240'}



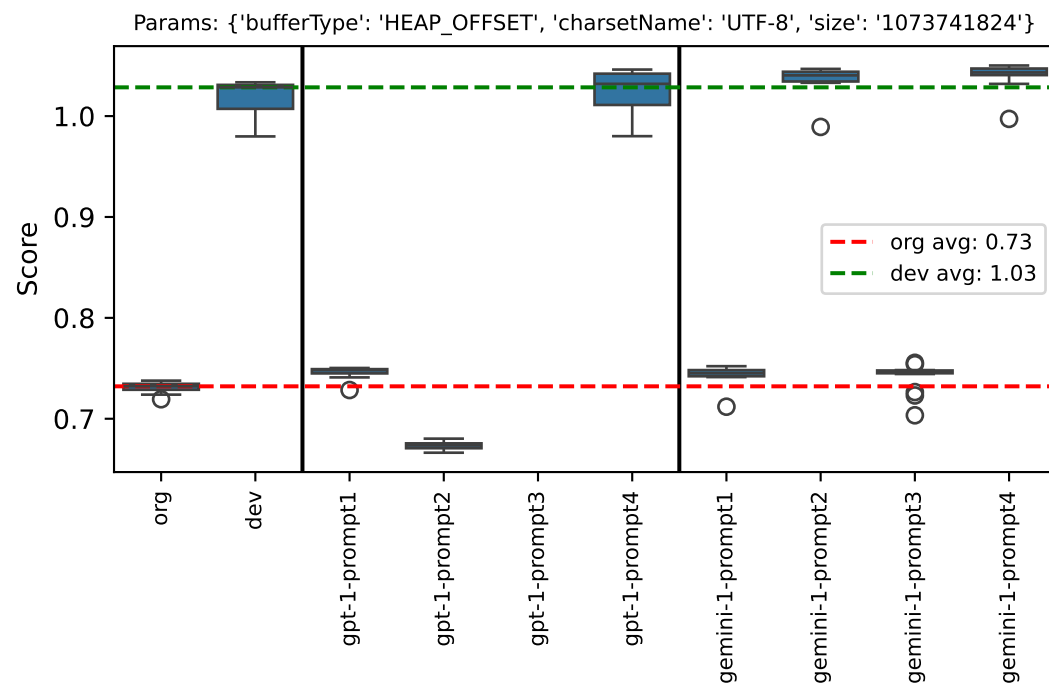
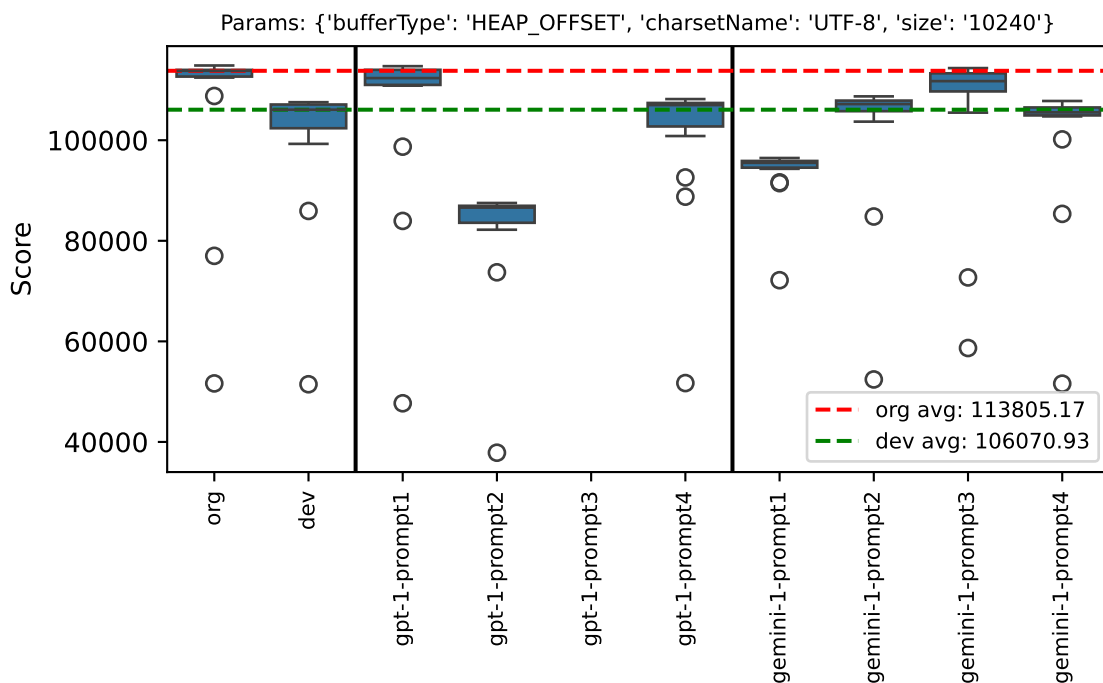
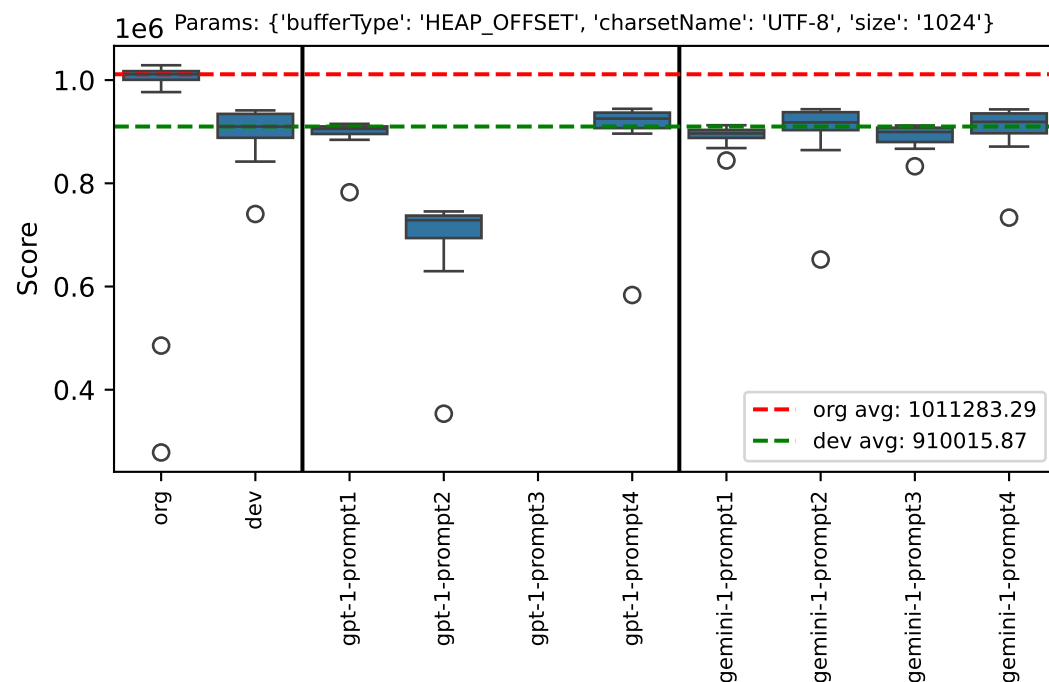
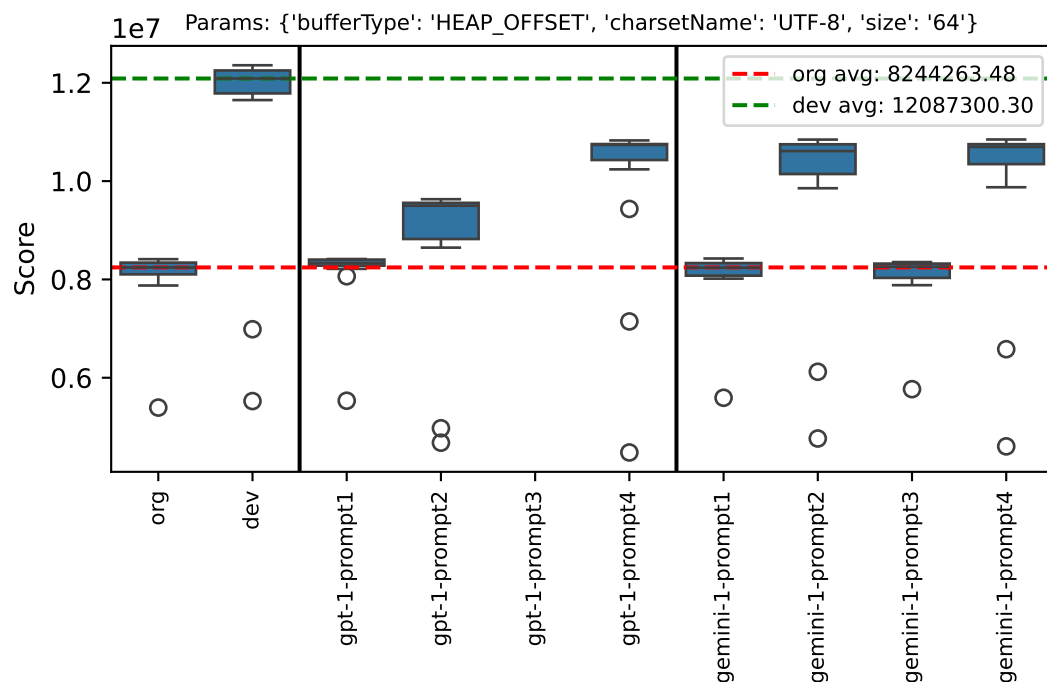
Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'US-ASCII', 'size': '1073741824'}



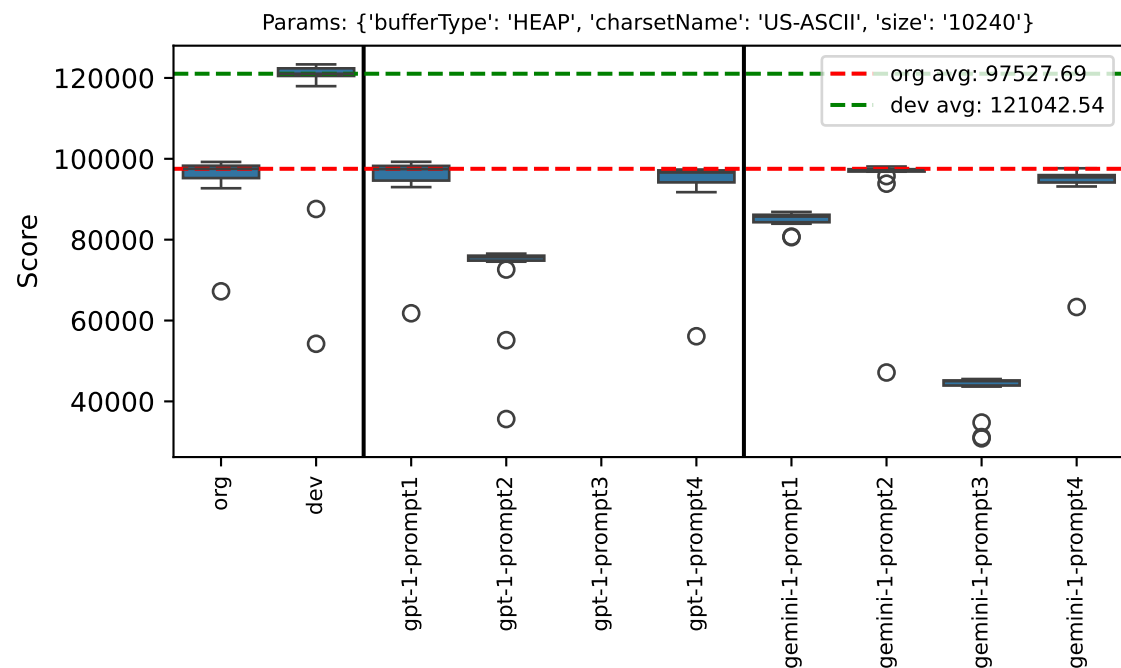
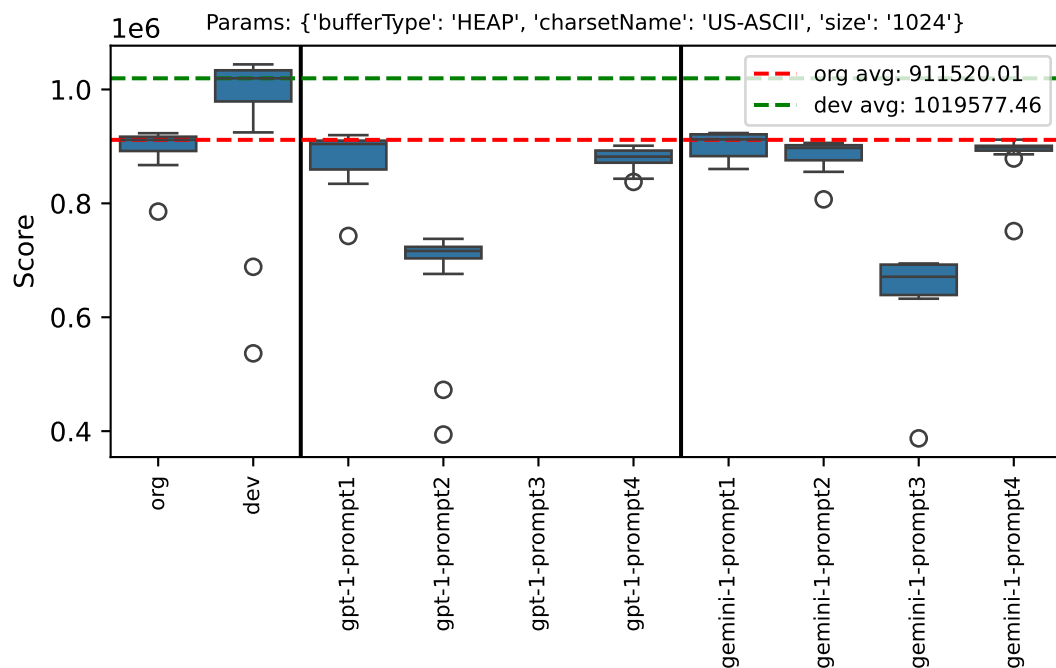
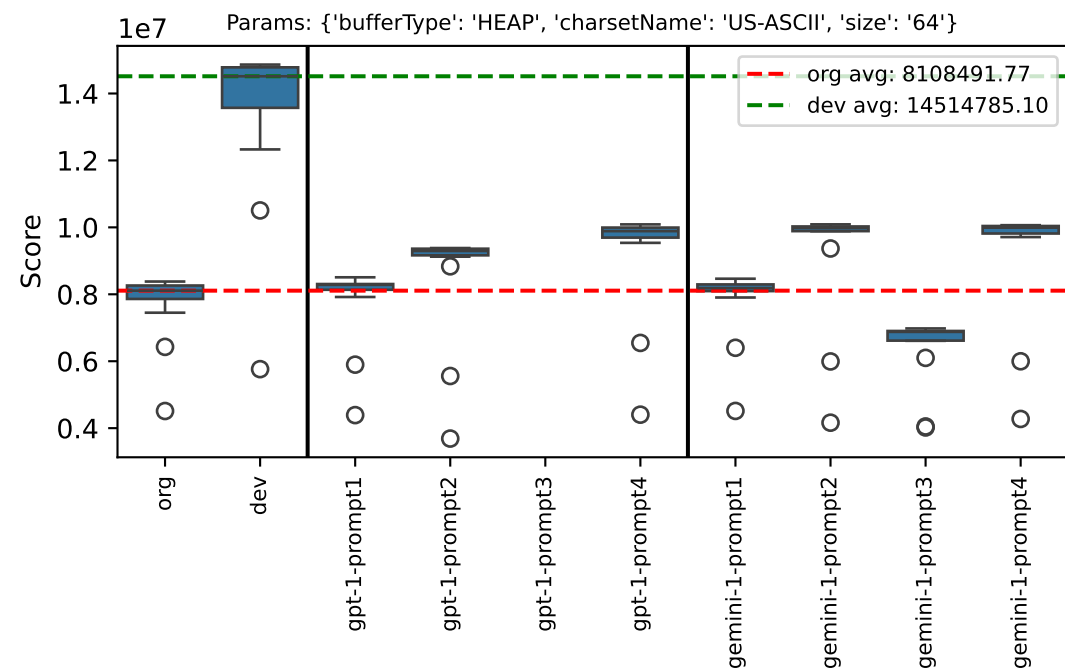
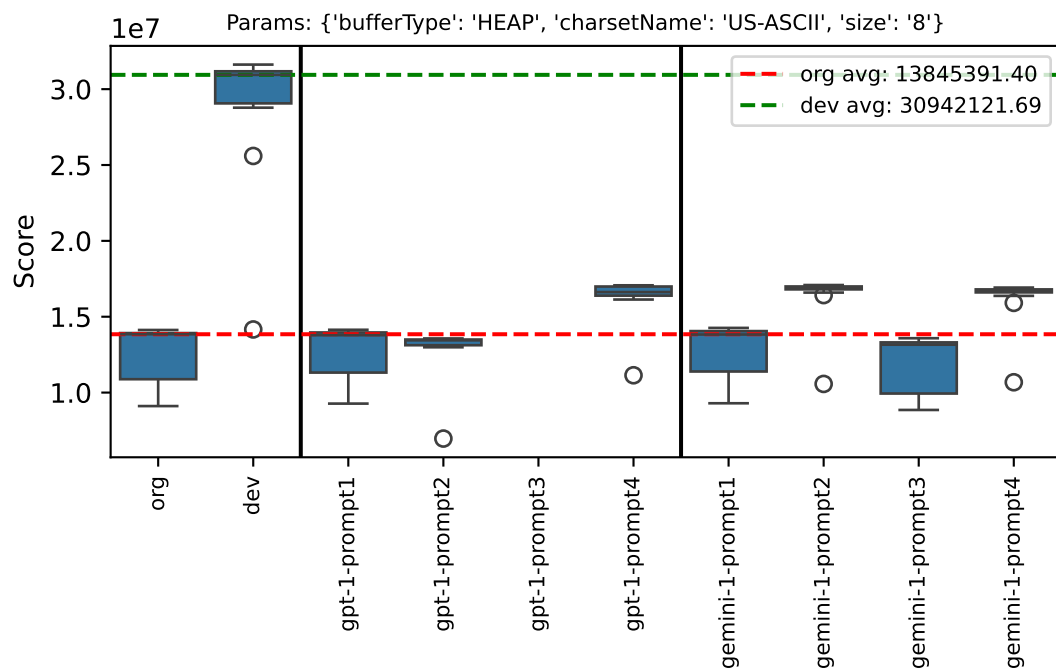
Params: {'bufferType': 'HEAP_OFFSET', 'charsetName': 'UTF-8', 'size': '8'}



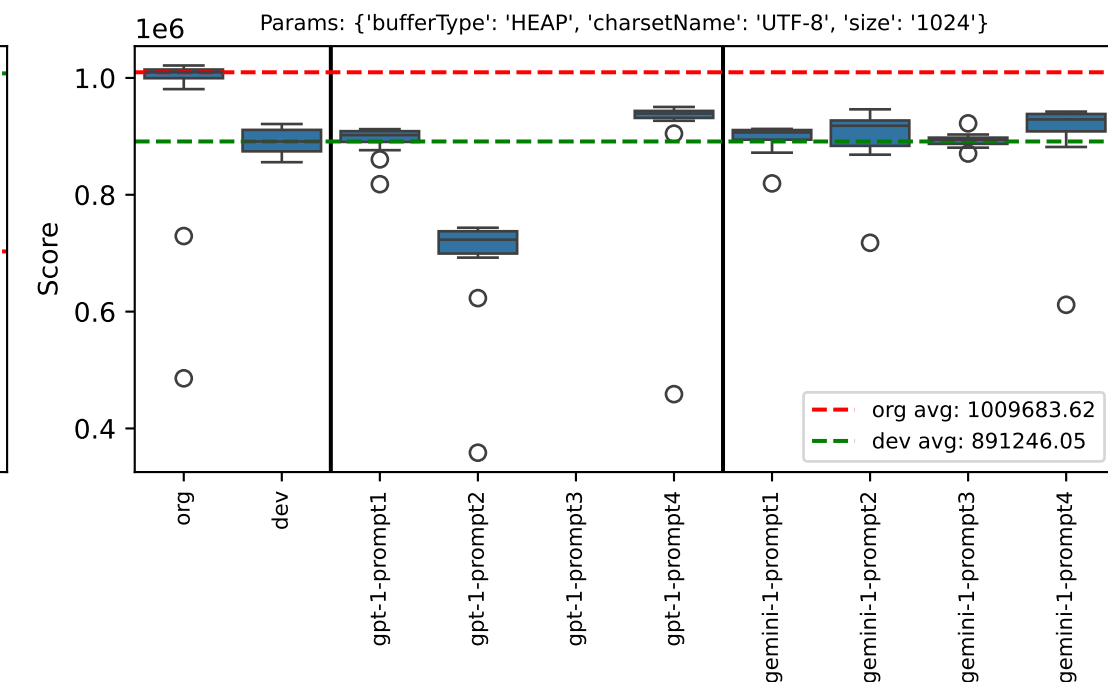
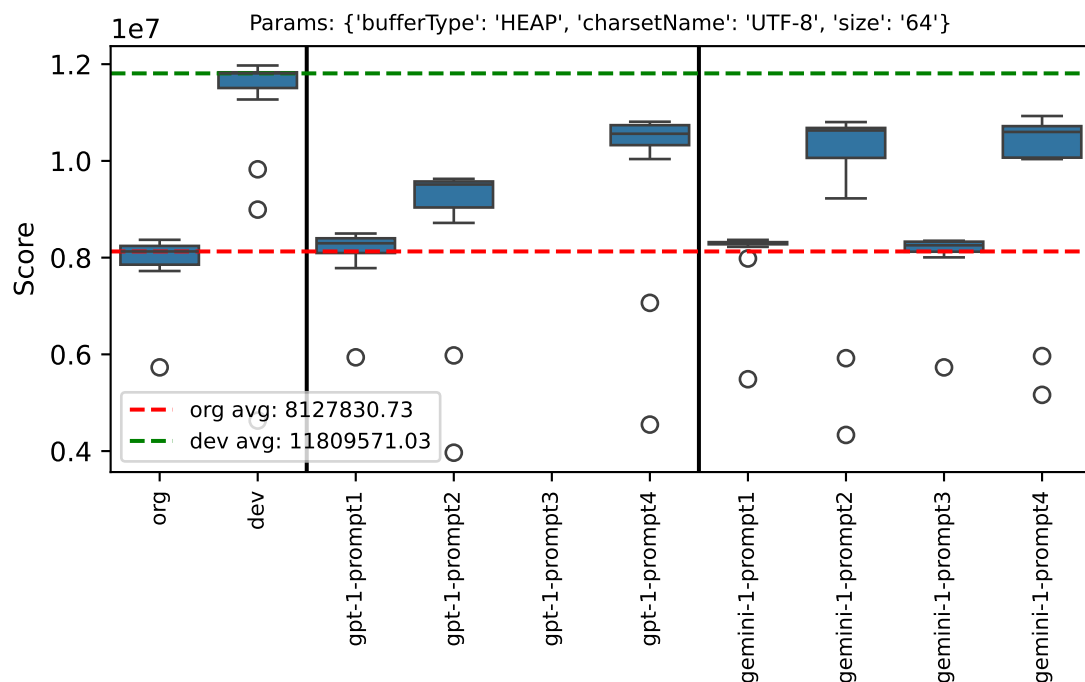
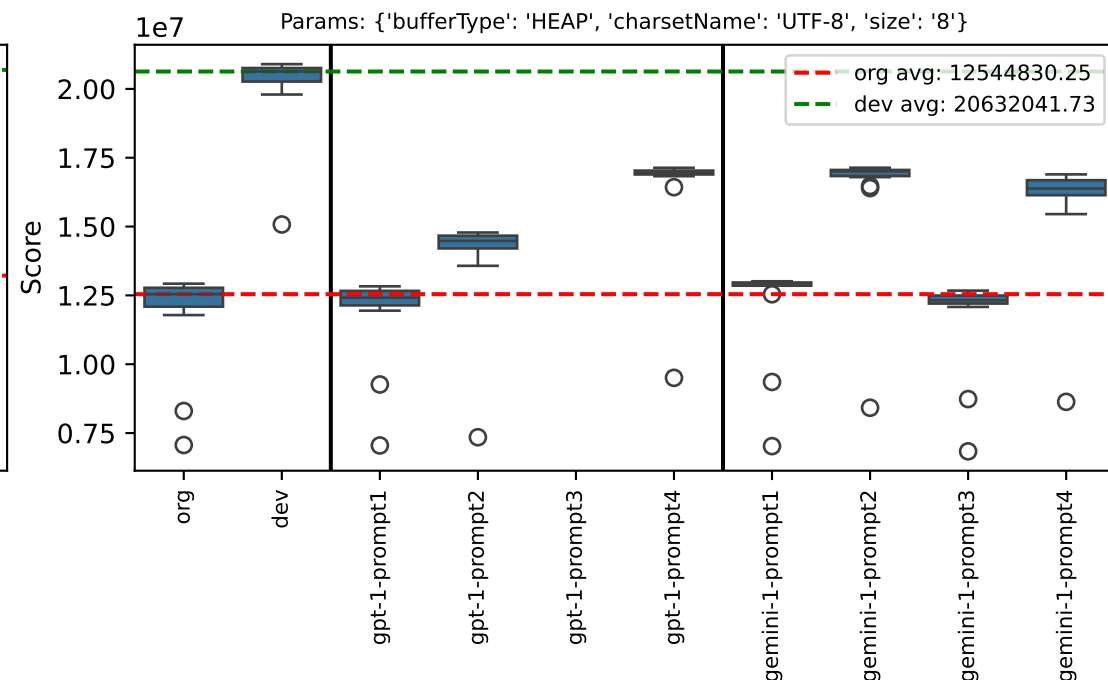
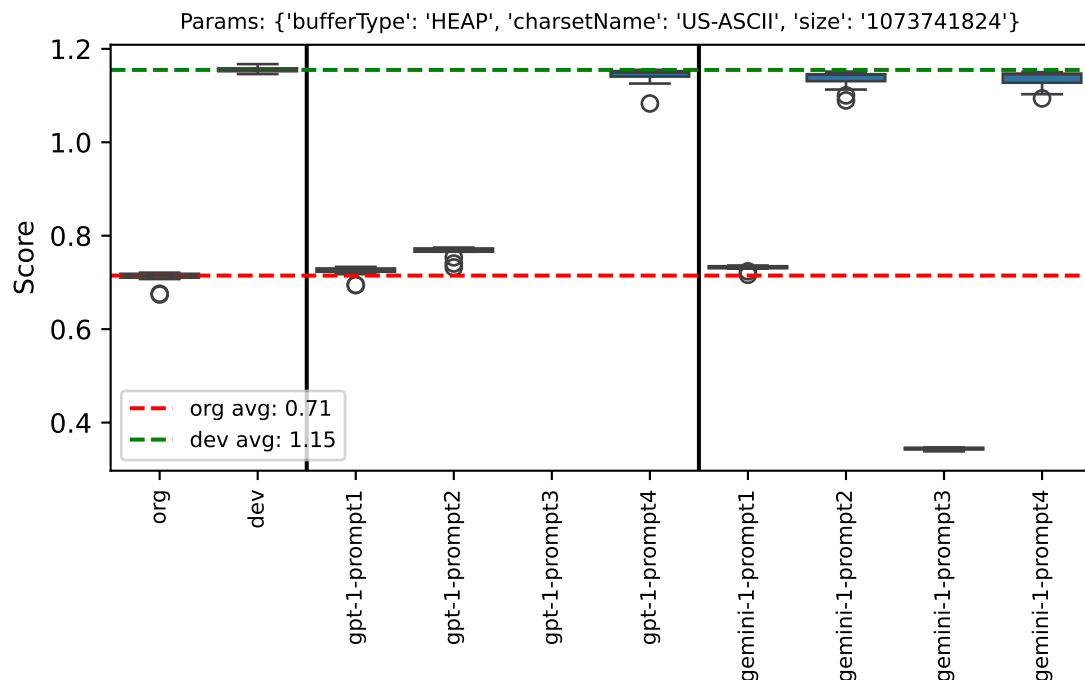
Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString



Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

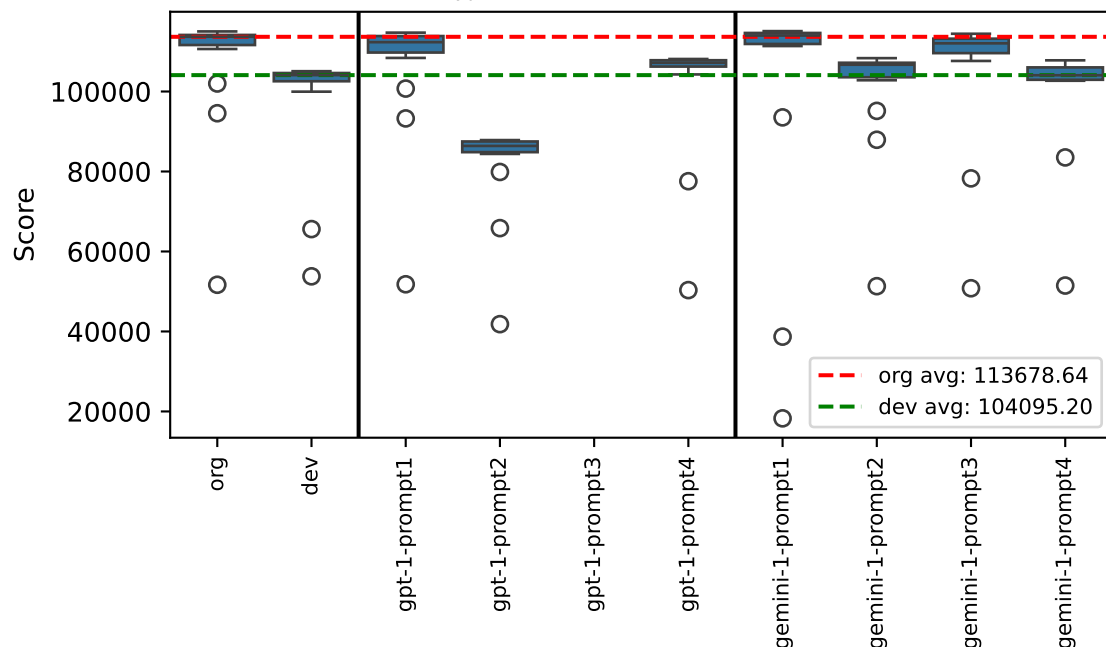


Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

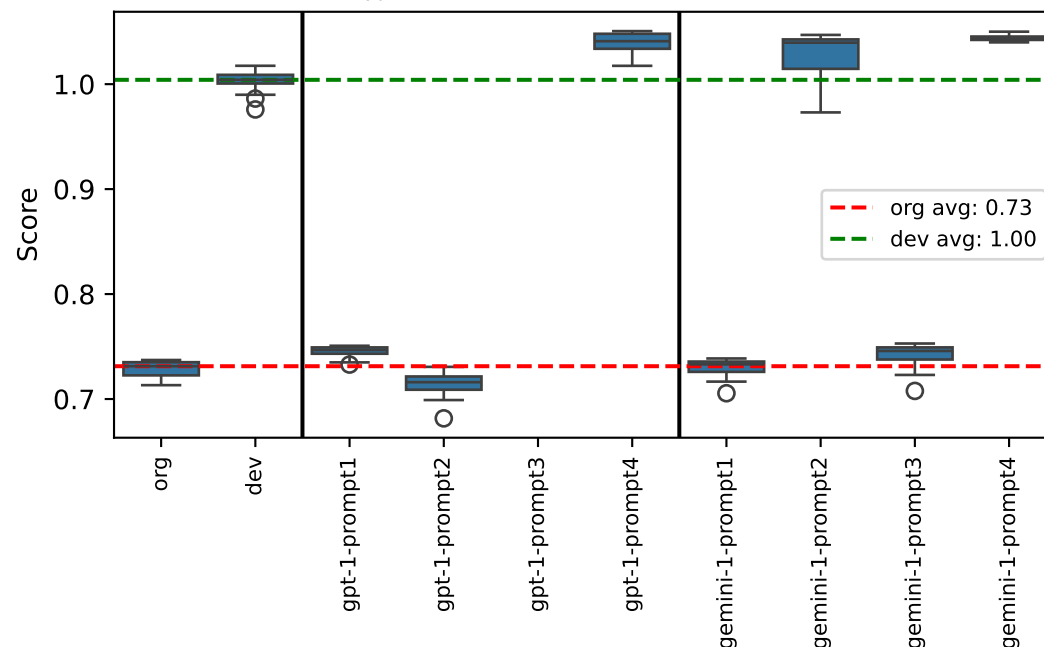


Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString

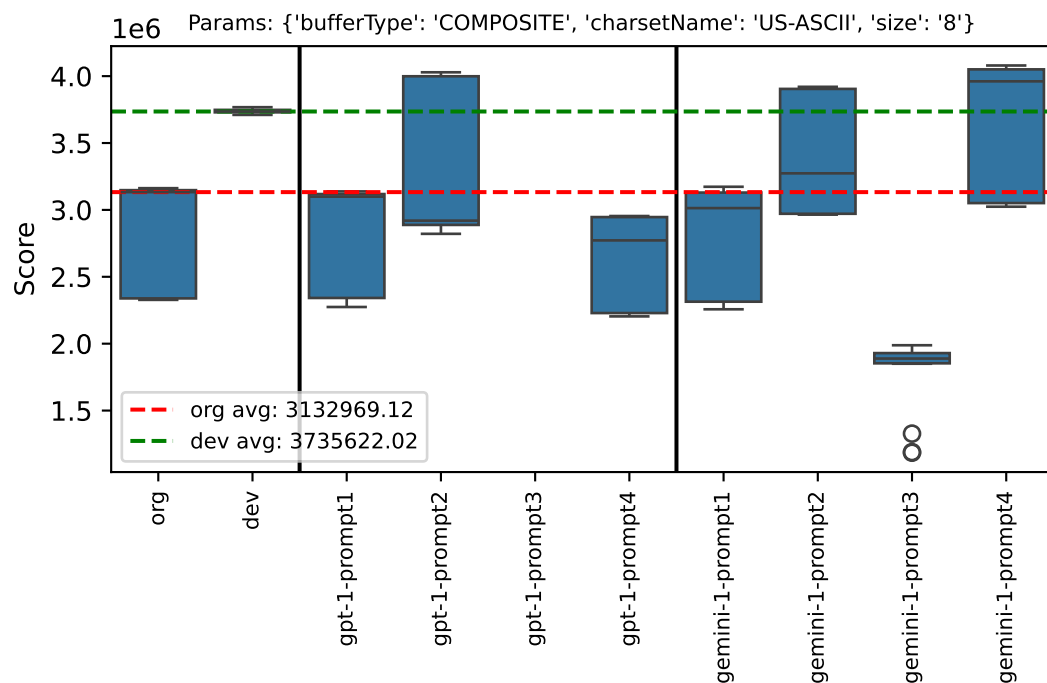
Params: {'bufferType': 'HEAP', 'charsetName': 'UTF-8', 'size': '10240'}



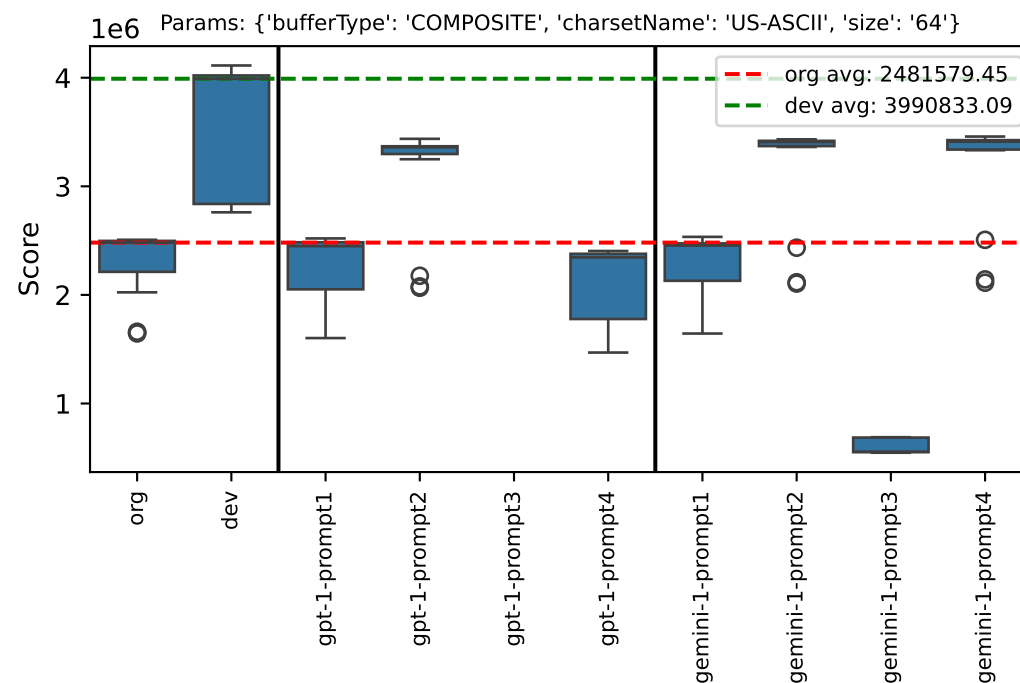
Params: {'bufferType': 'HEAP', 'charsetName': 'UTF-8', 'size': '1073741824'}



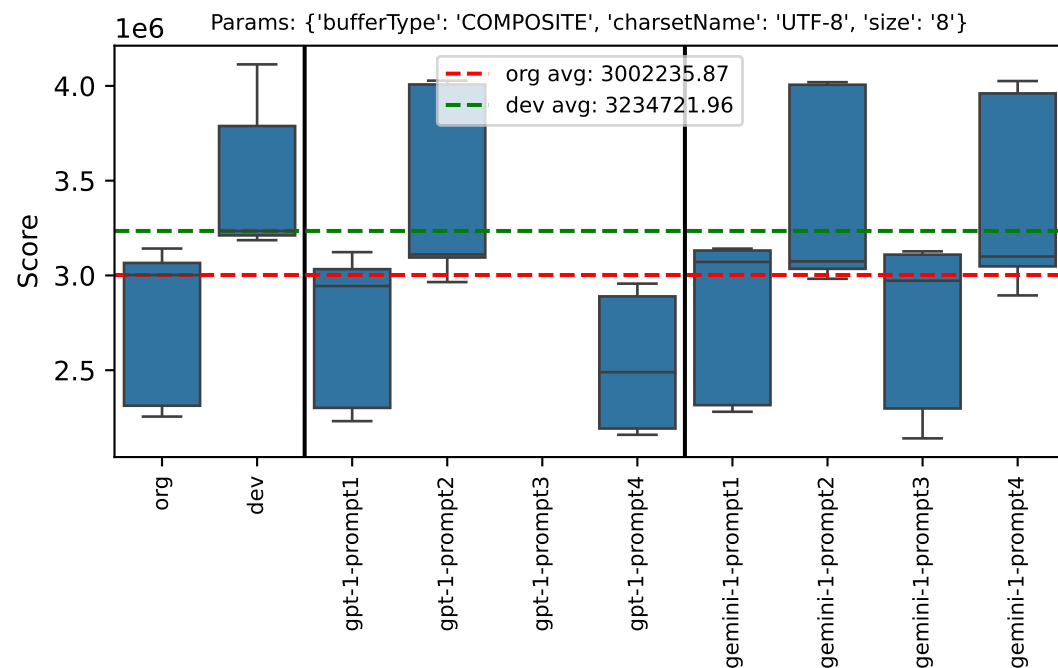
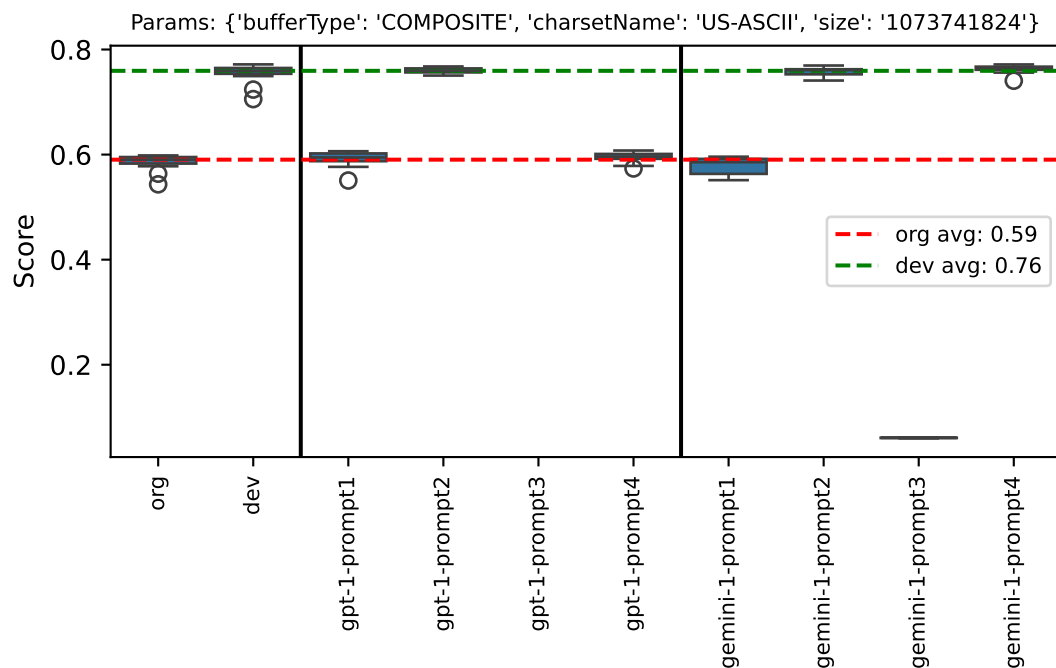
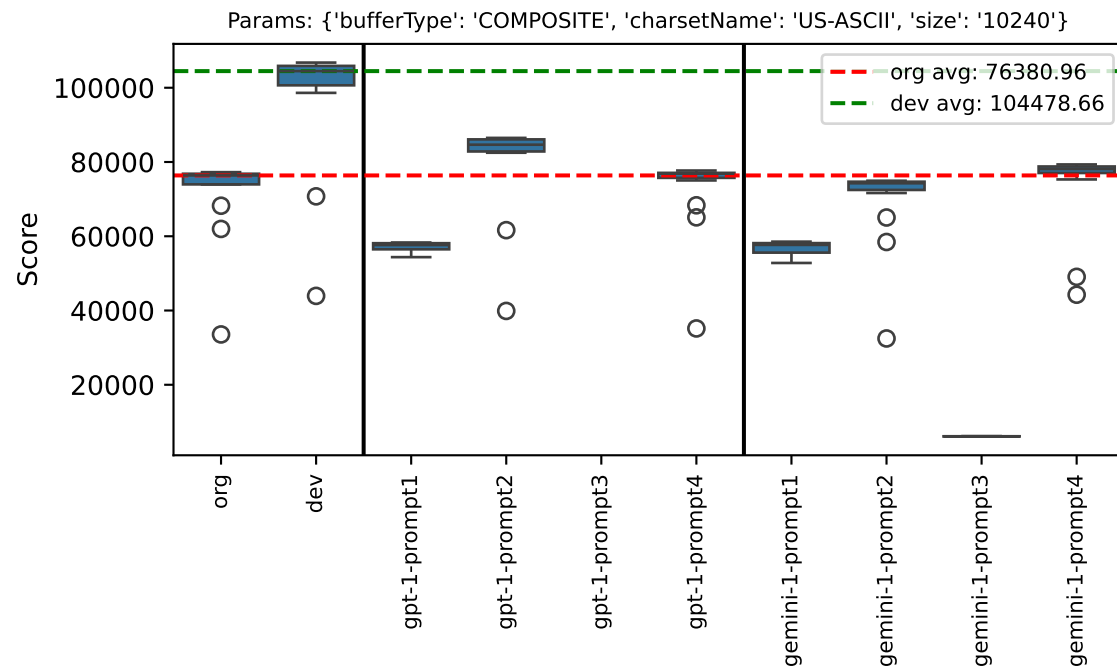
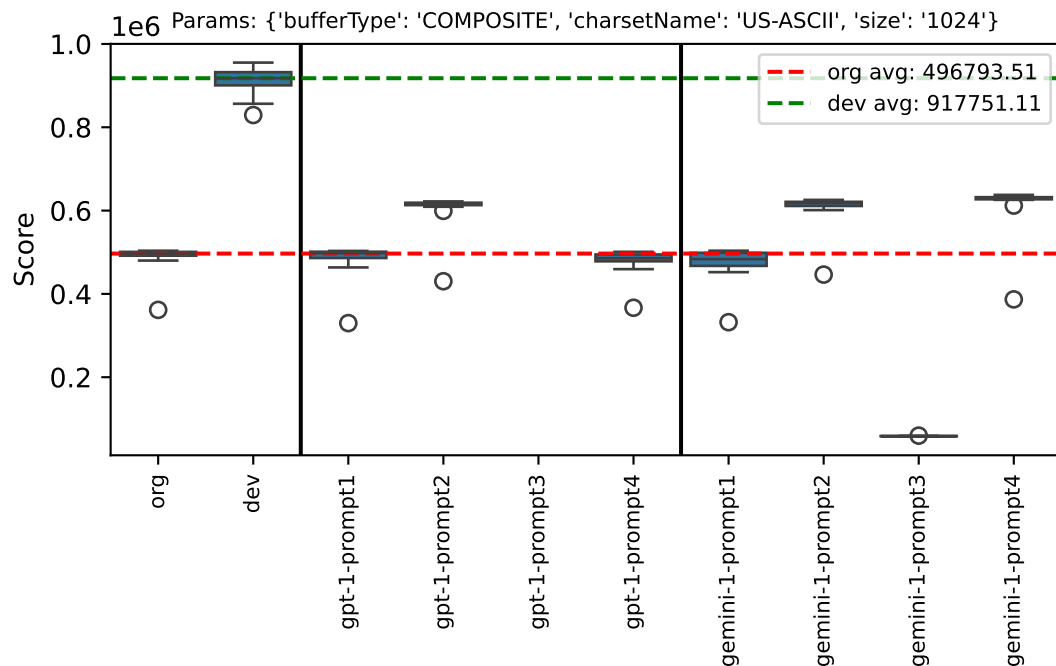
Params: {'bufferType': 'COMPOSITE', 'charsetName': 'US-ASCII', 'size': '8'}



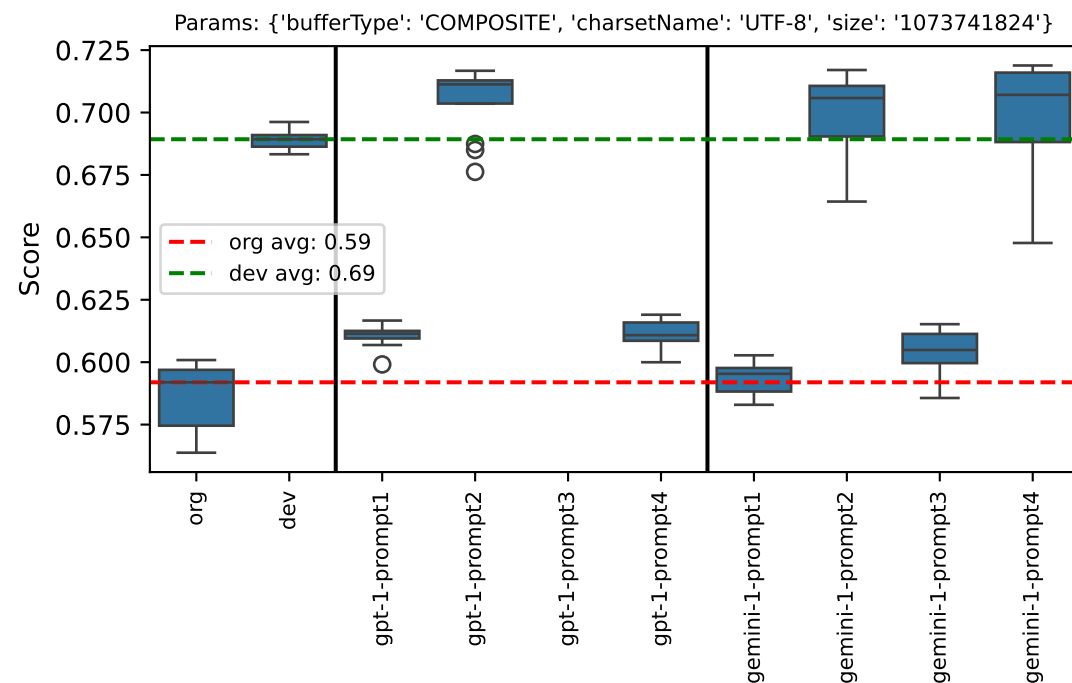
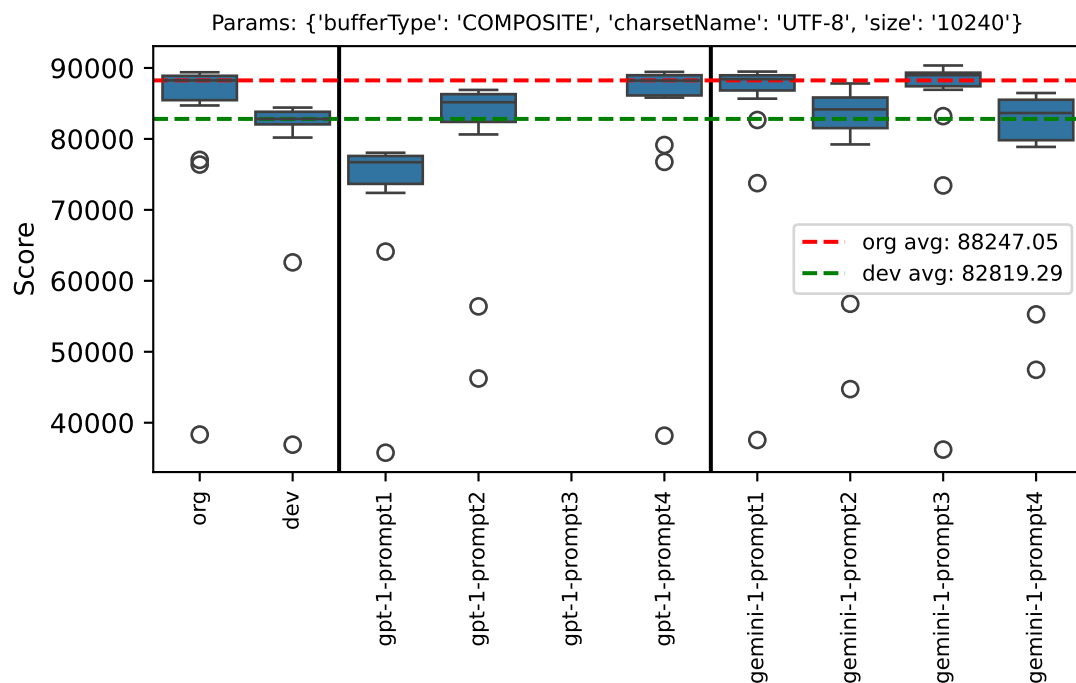
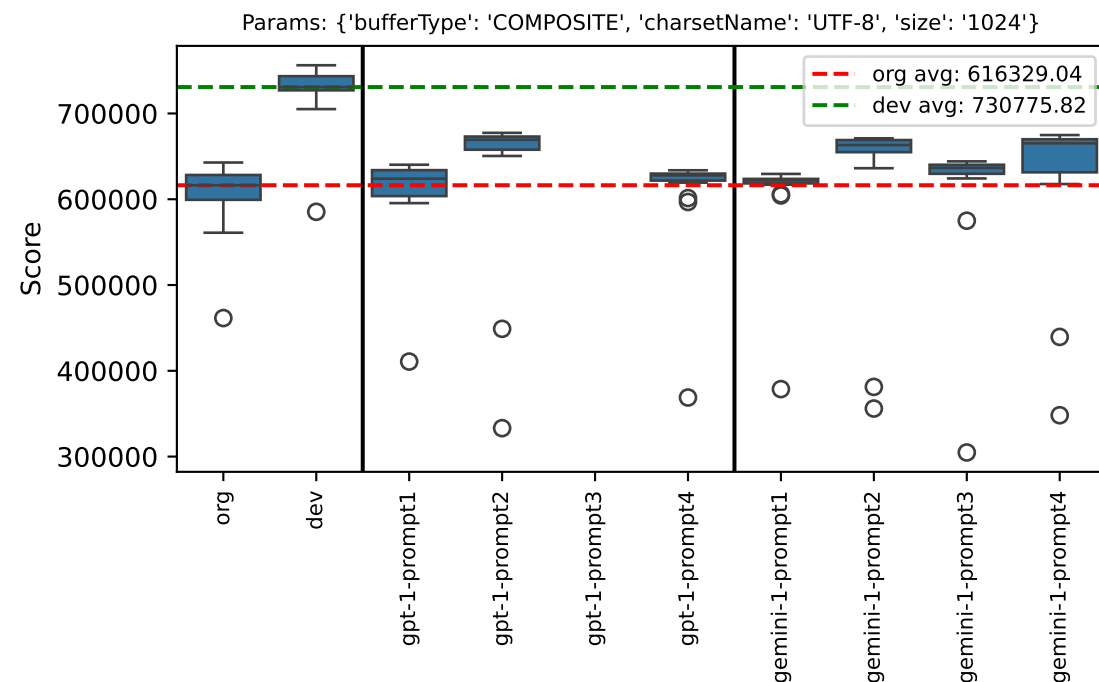
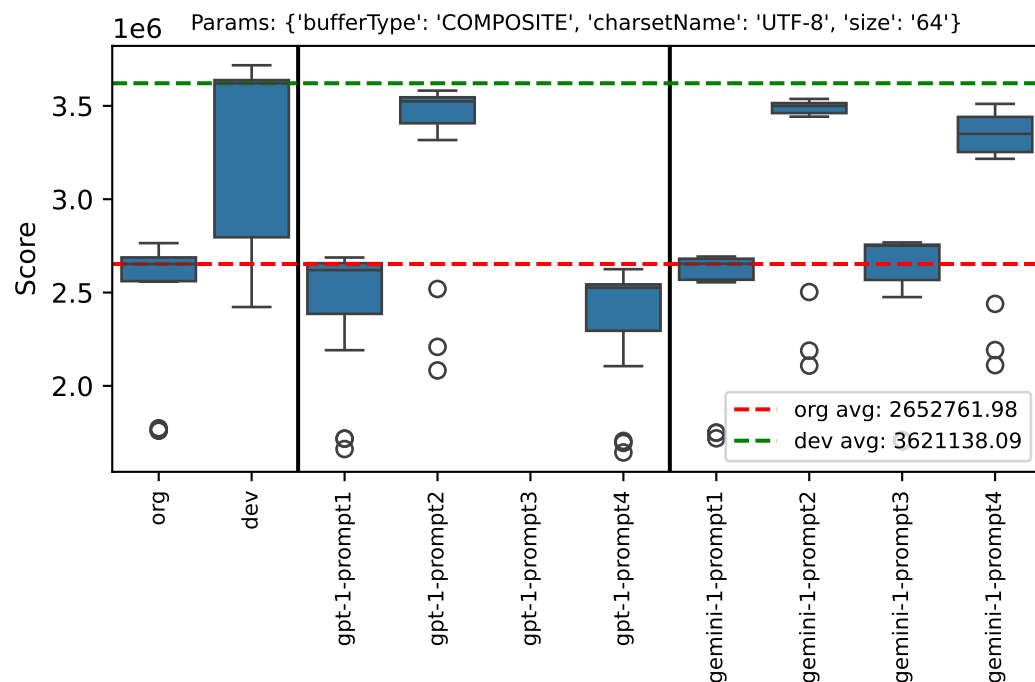
Params: {'bufferType': 'COMPOSITE', 'charsetName': 'US-ASCII', 'size': '64'}



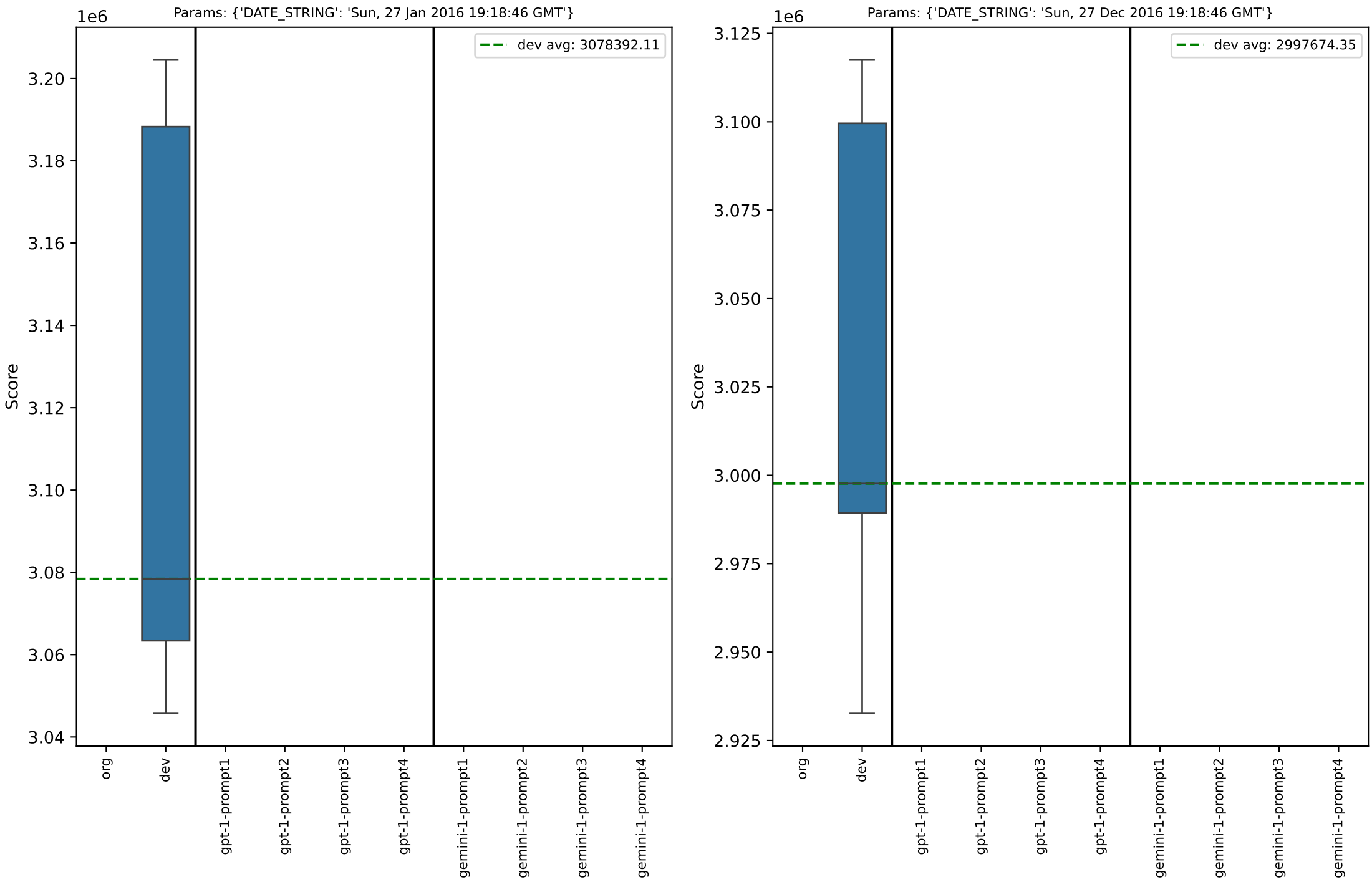
Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString



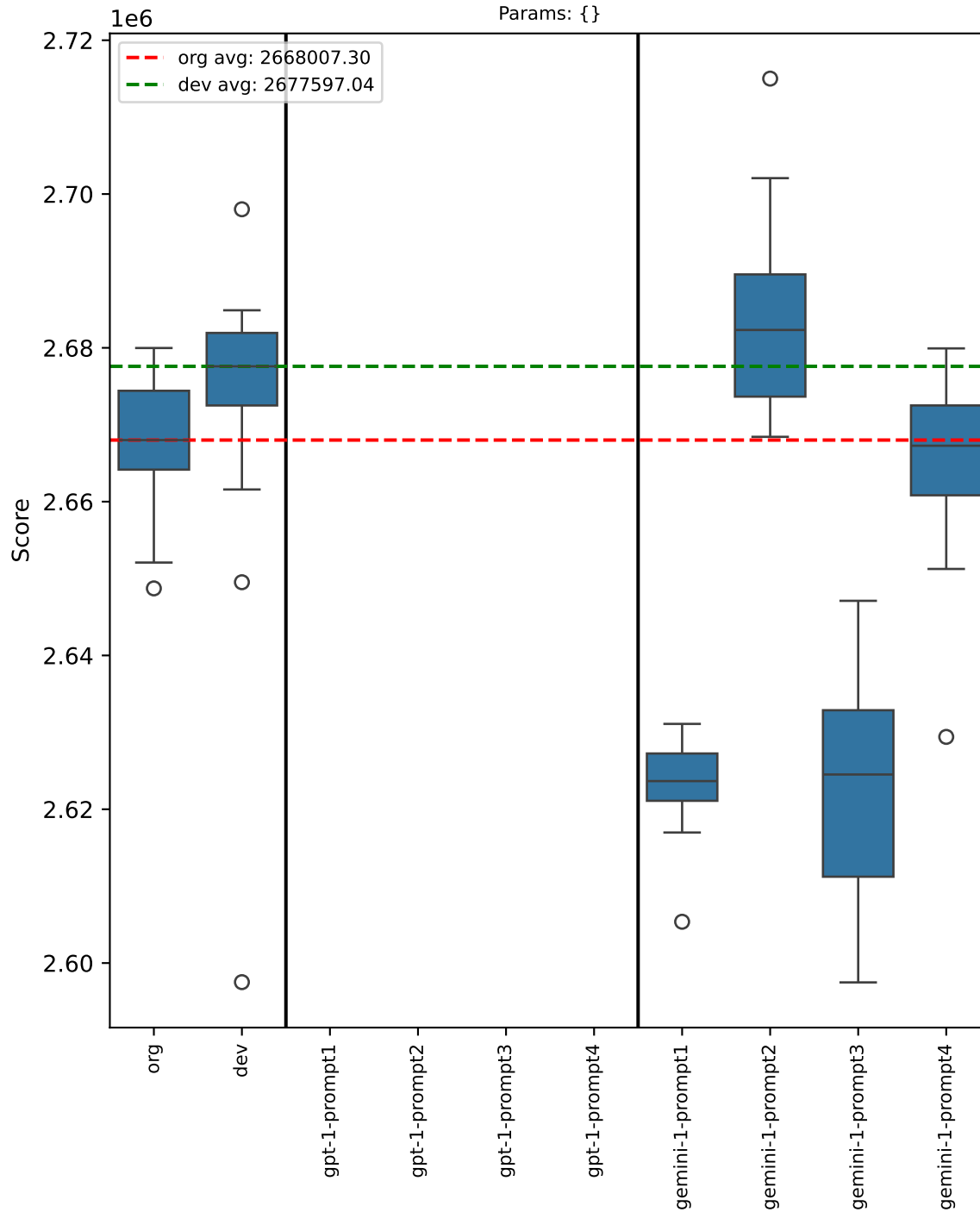
Benchmark: io.netty.buffer.ByteBufUtilDecodeStringBenchmark.decodeString



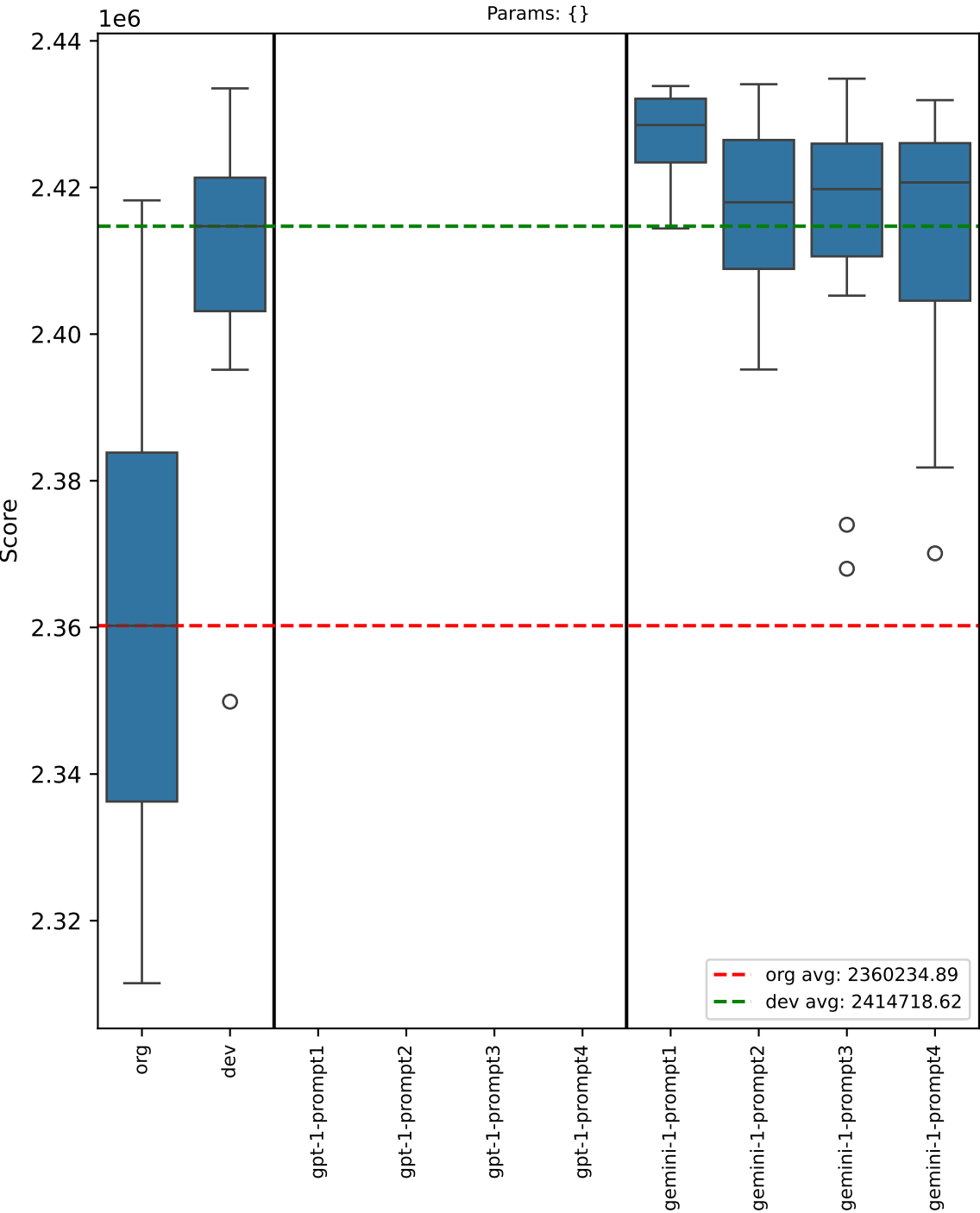
Benchmark: io.netty.handler.codec.DateFormatter2Benchmark.parseHttpHeaderDateFormatterNew



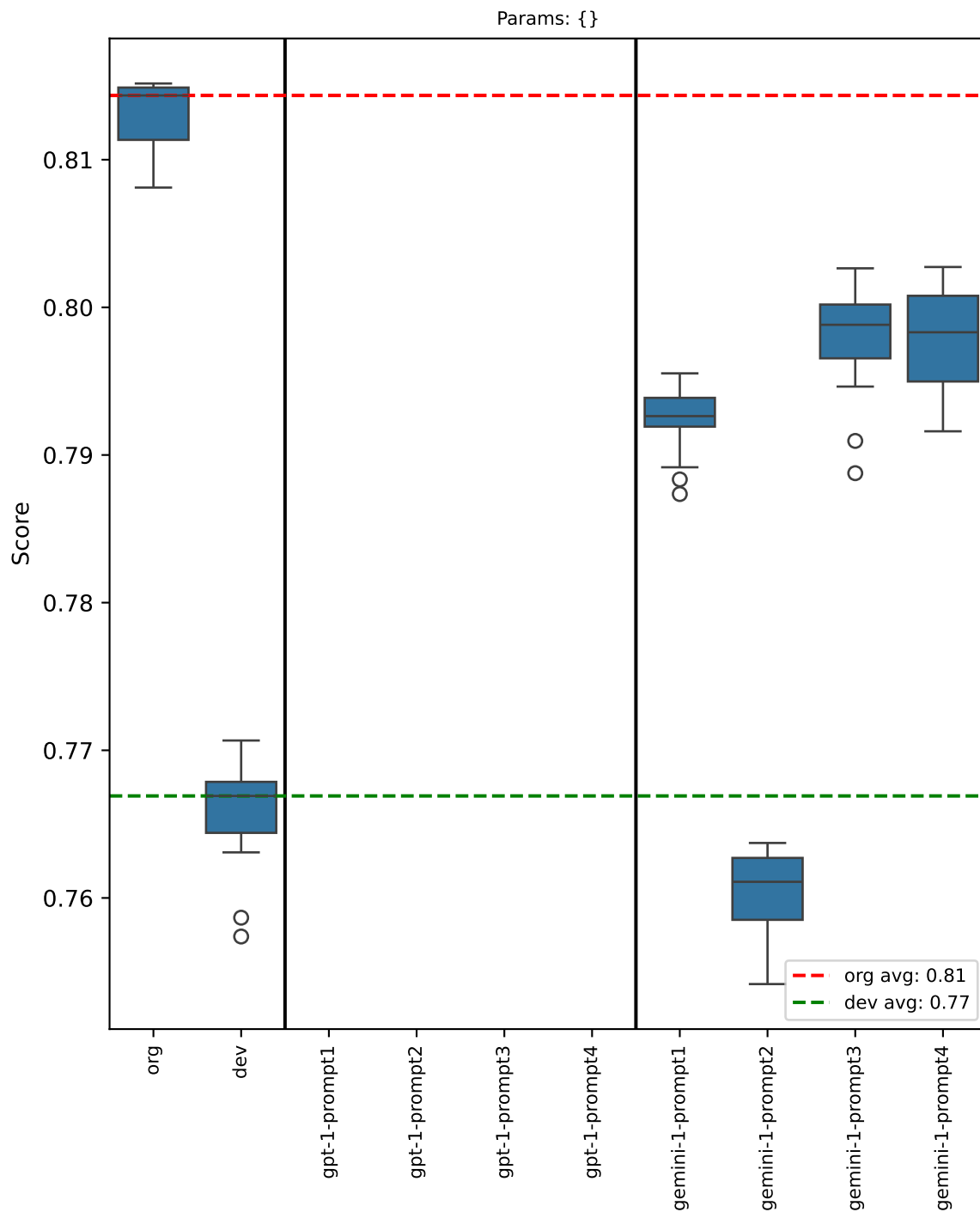
Benchmark: io.netty.handler.codec.http.HttpRequestEncoderInsertBenchmark.newEncoder



Benchmark: io.netty.handler.codec.http.HttpRequestEncoderInsertBenchmark.oldEncoder

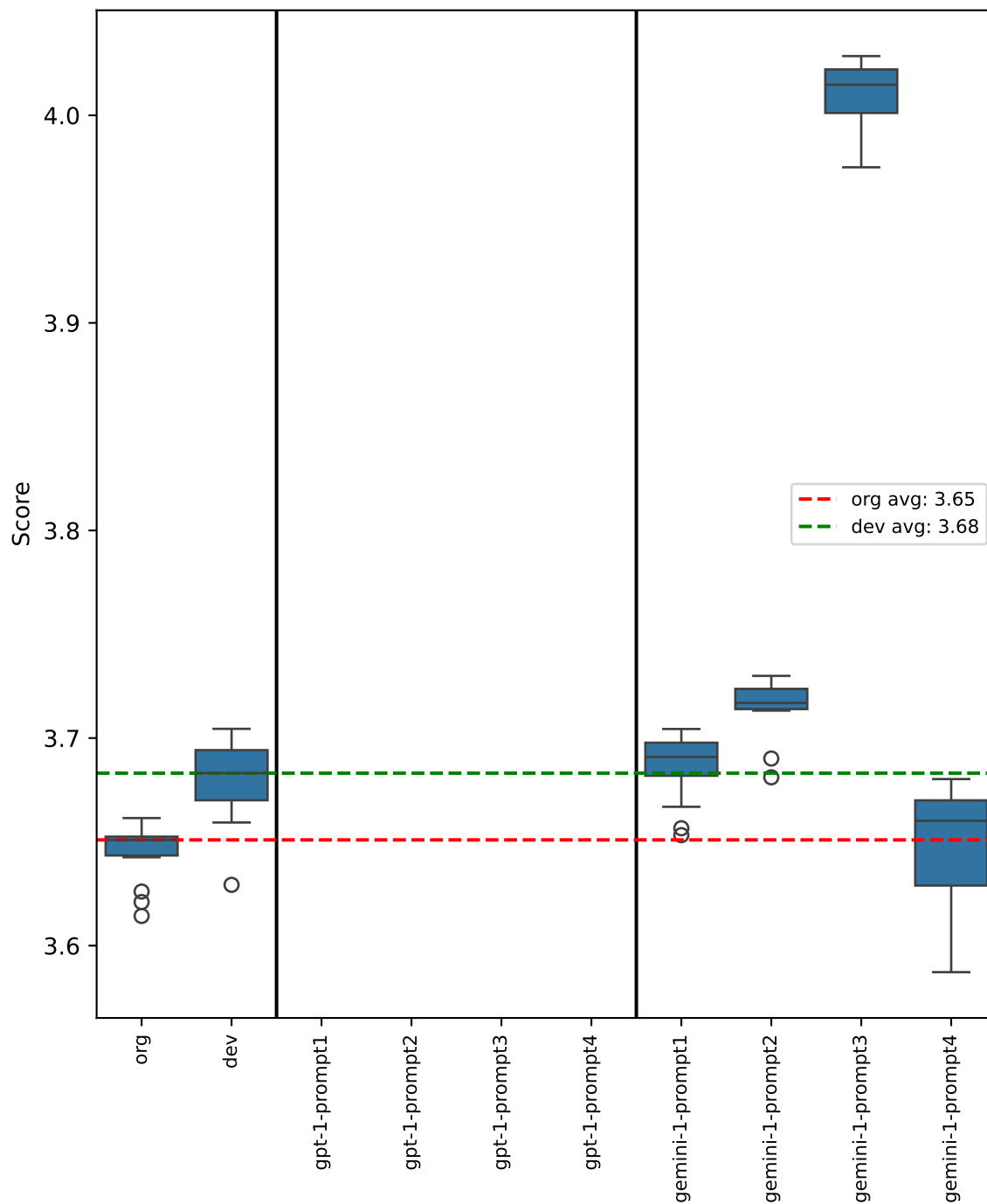


Benchmark: io.netty.handler.codec.http.QueryStringDecoderBenchmark.mixedDecoding



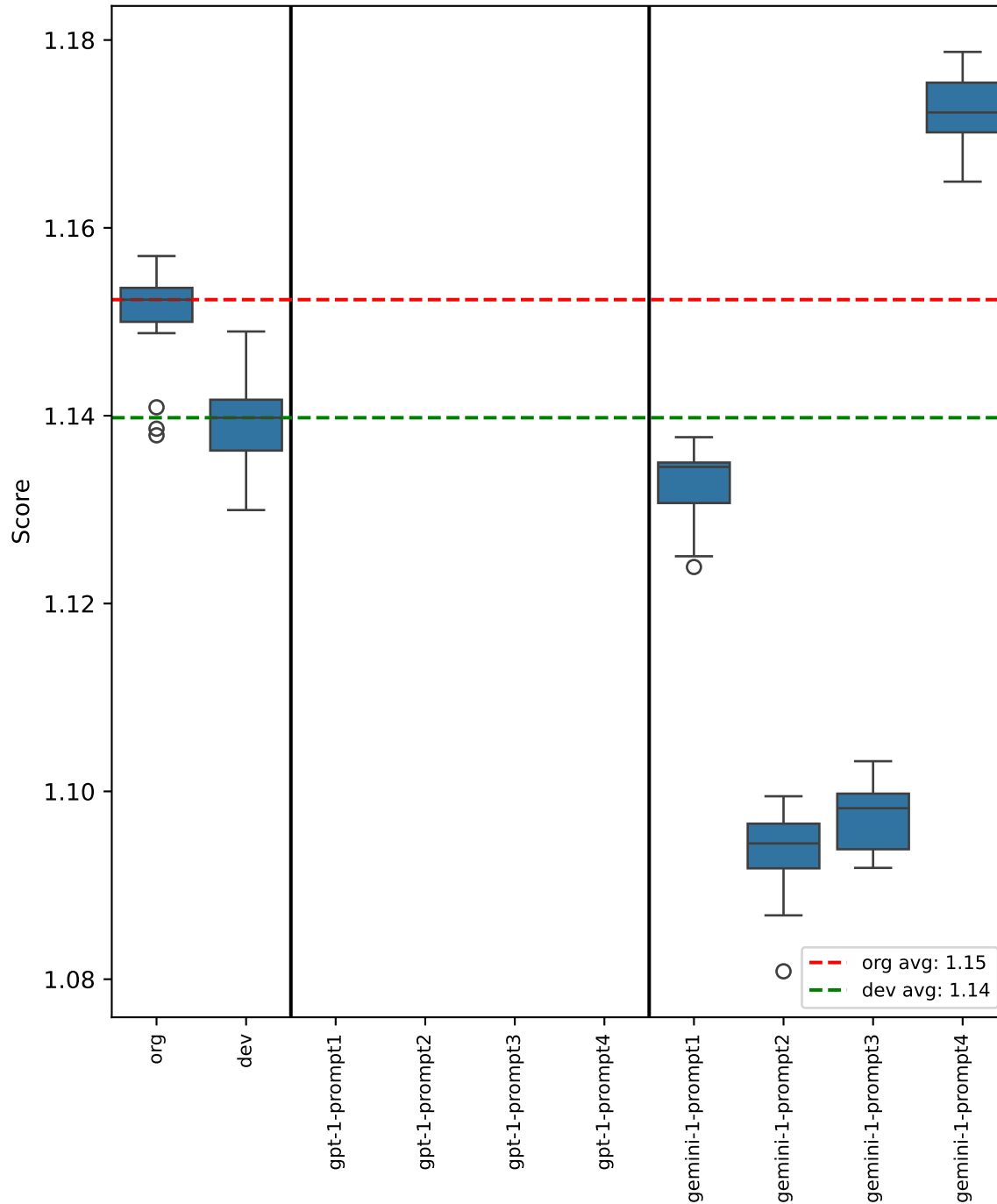
Benchmark: io.netty.handler.codec.http.QueryStringDecoderBenchmark.noDecoding

Params: {}

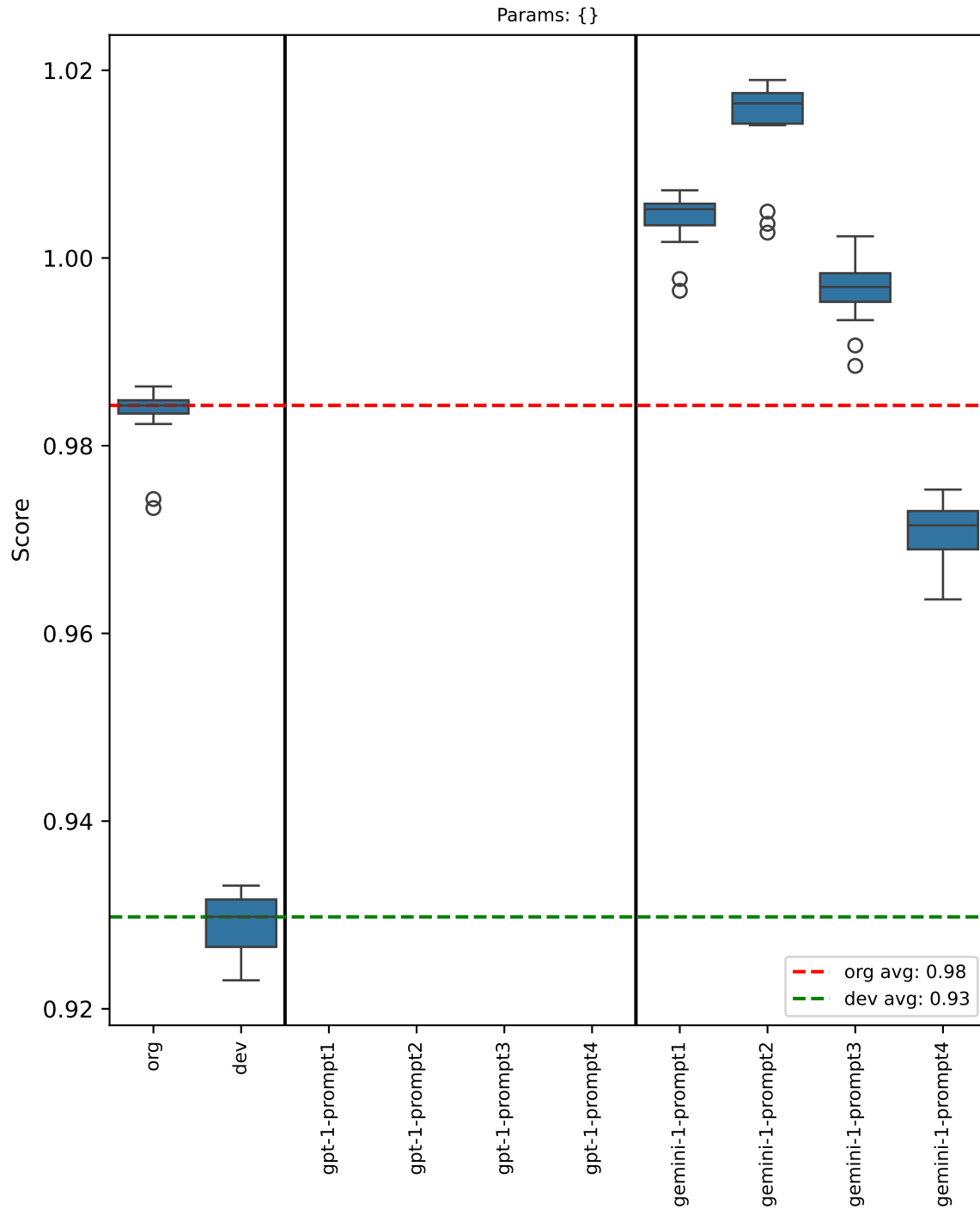


Benchmark: io.netty.handler.codec.http.QueryStringDecoderBenchmark.nonStandardDecoding

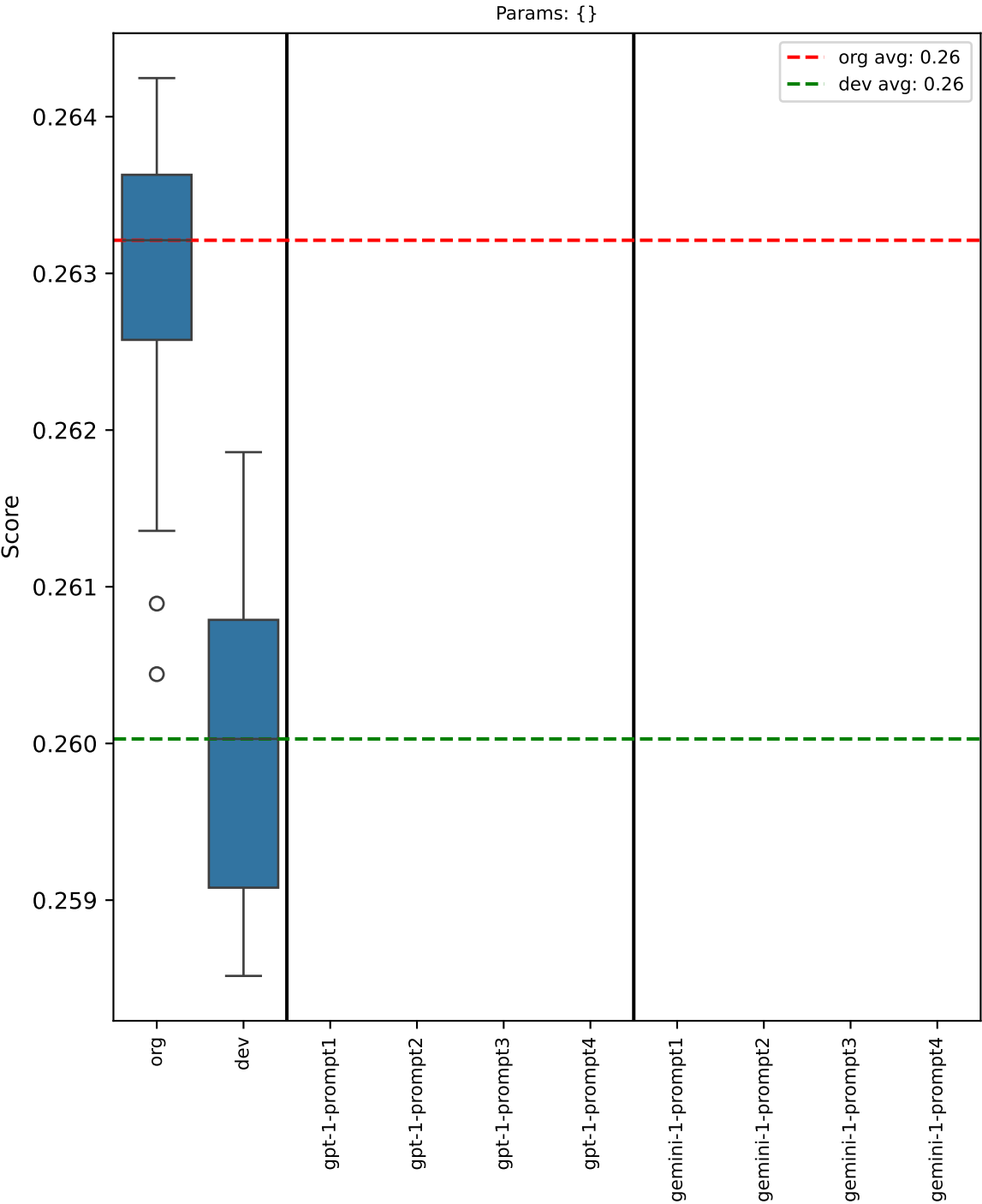
Params: {}



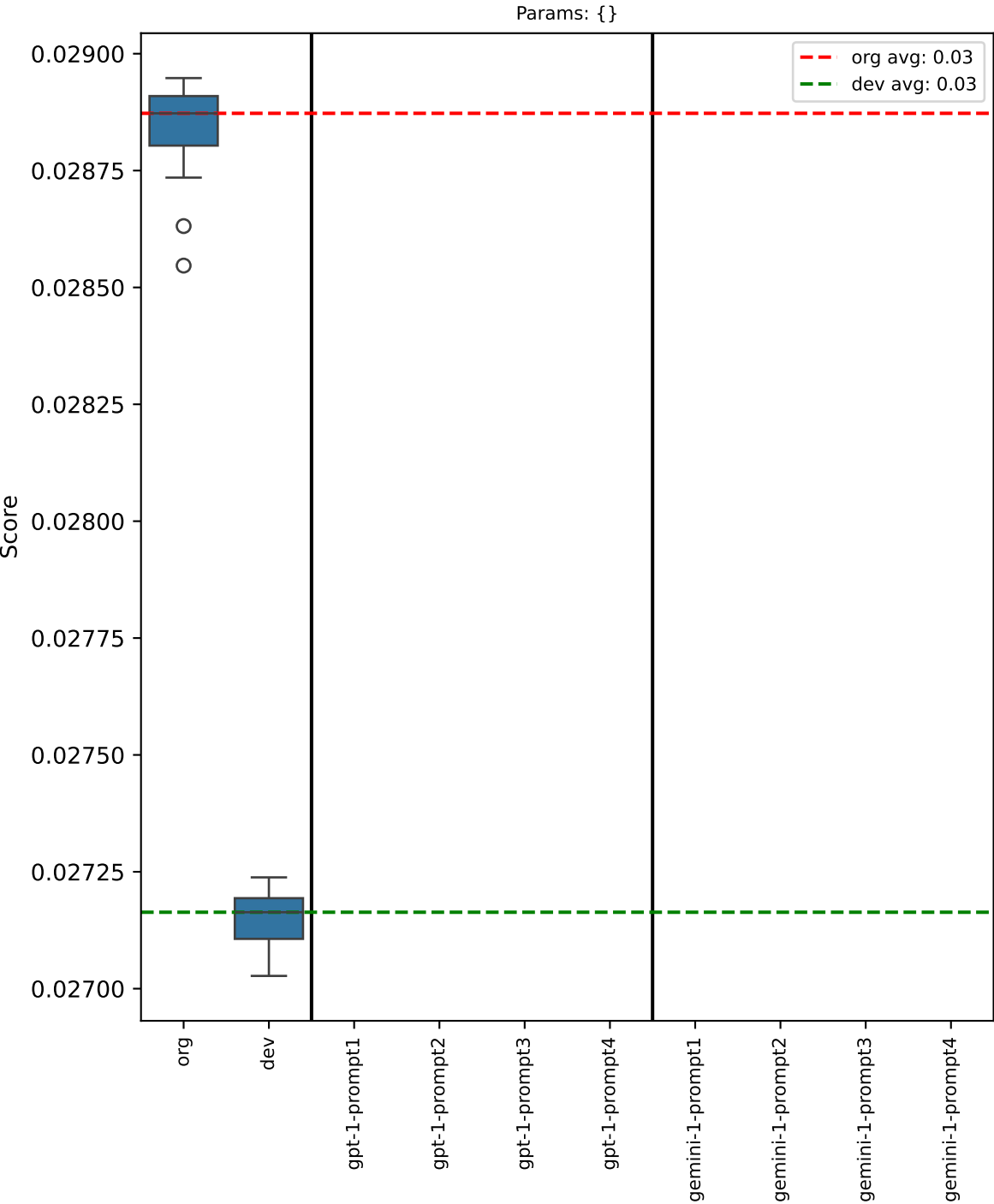
Benchmark: io.netty.handler.codec.http.QueryStringDecoderBenchmark.onlyDecoding



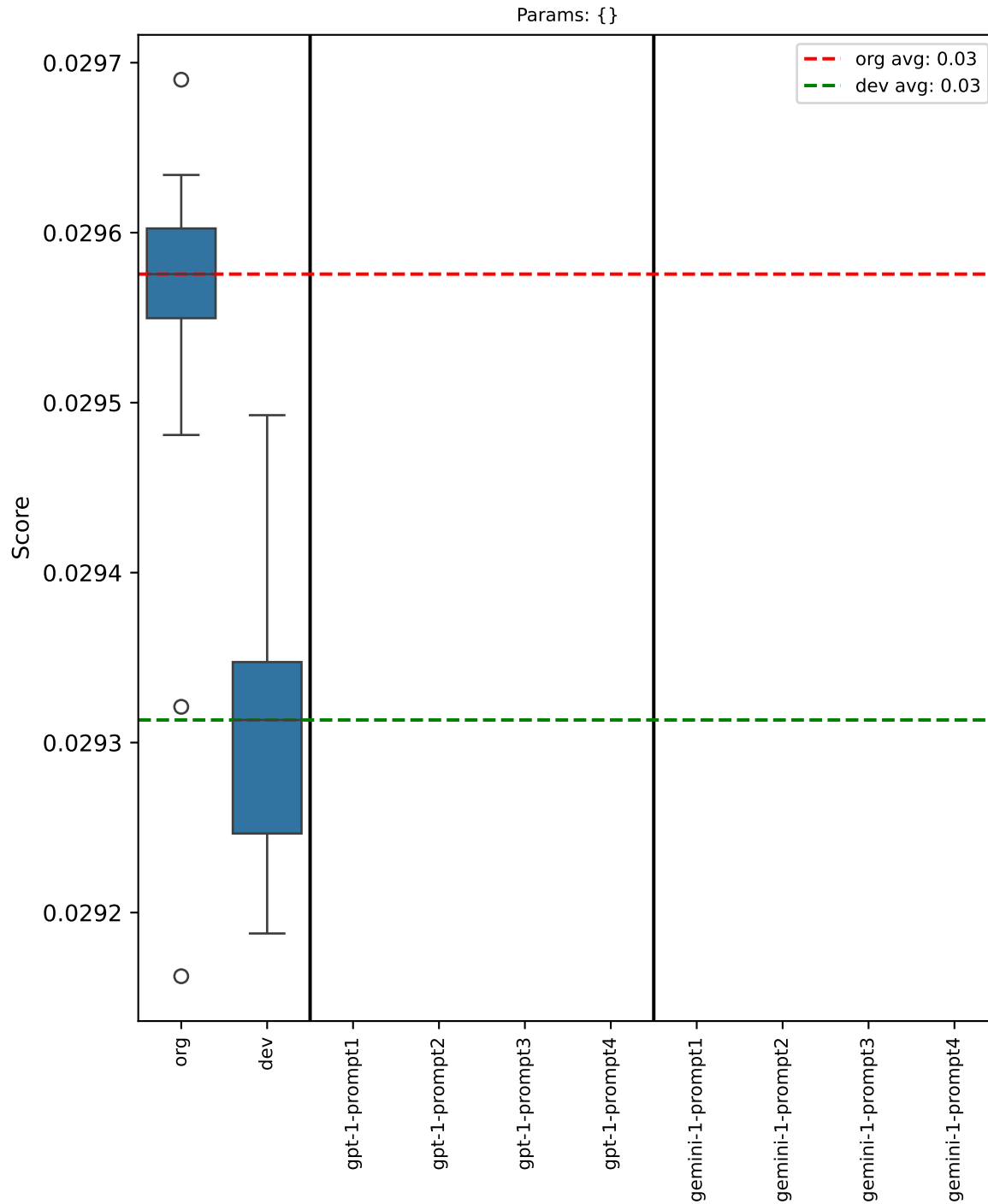
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.longAscii



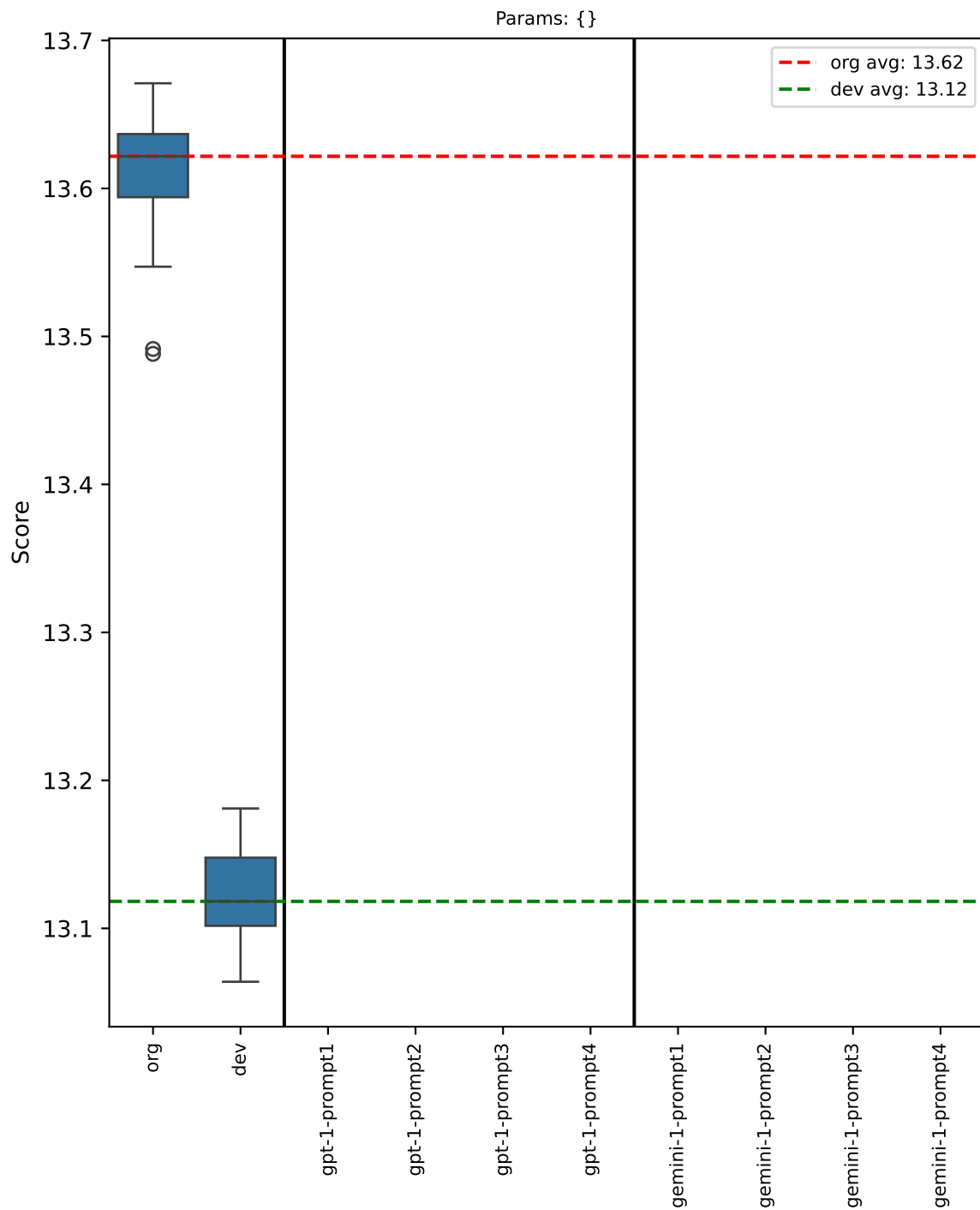
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.longAsciiFirst



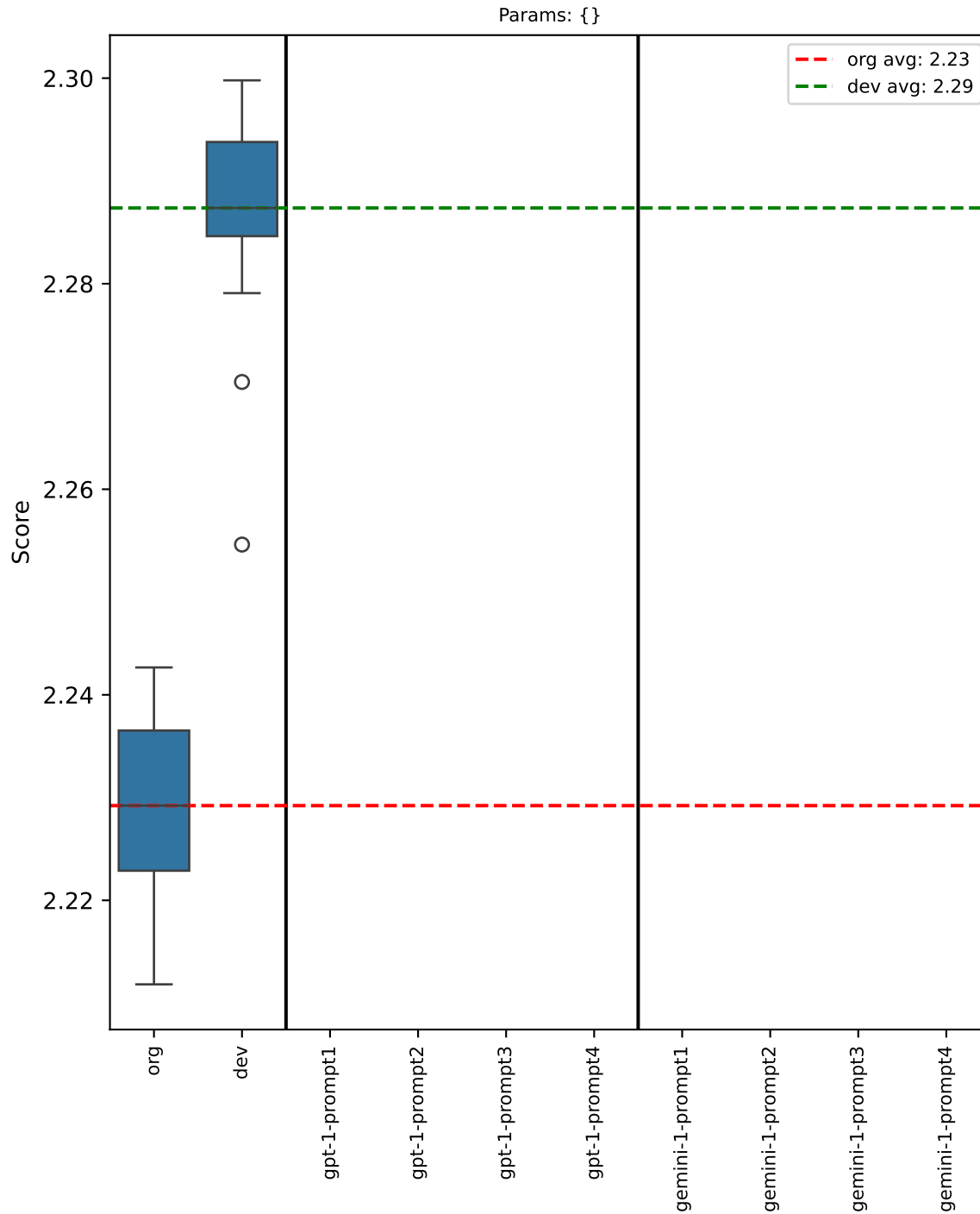
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.longUtf8



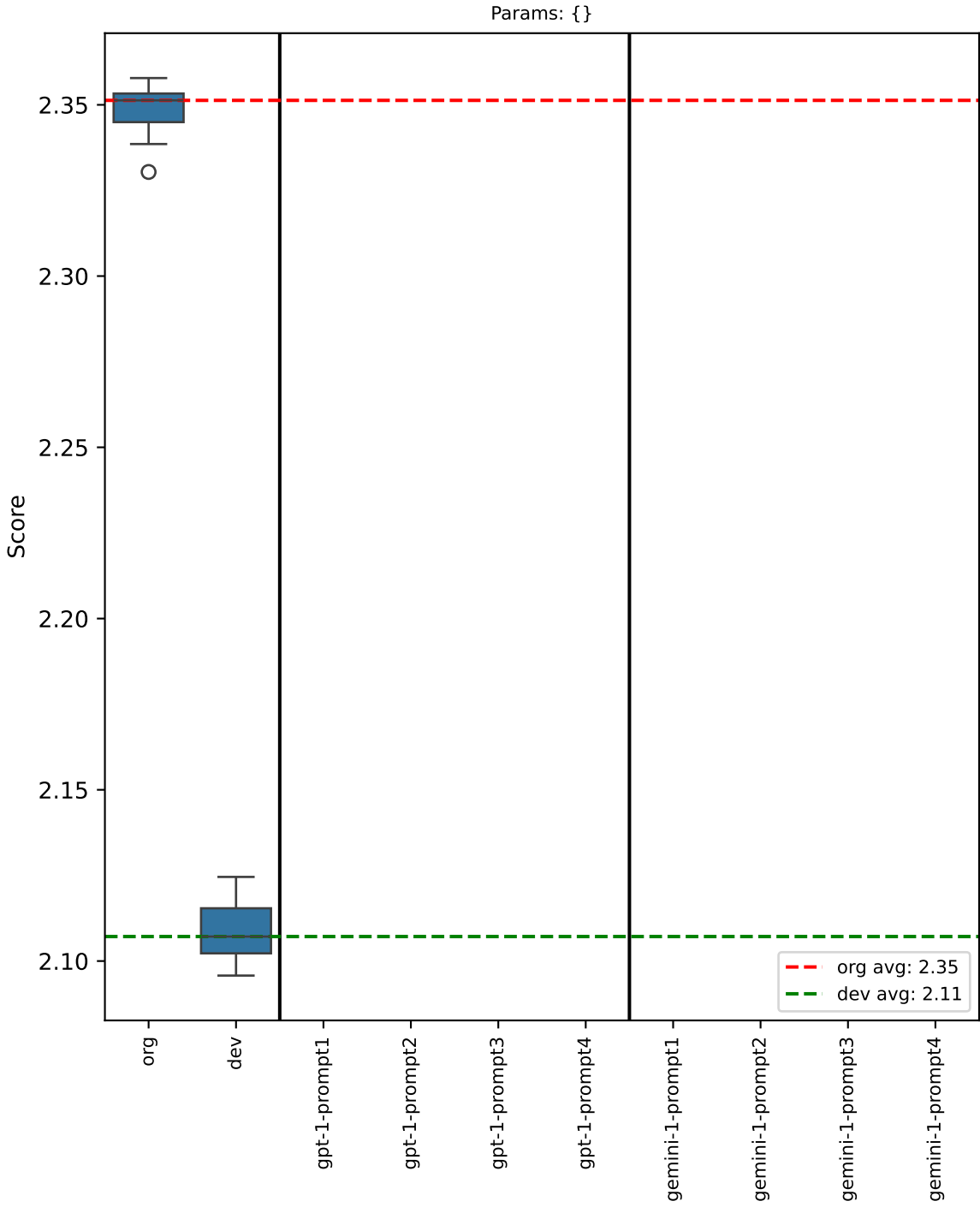
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.shortAscii



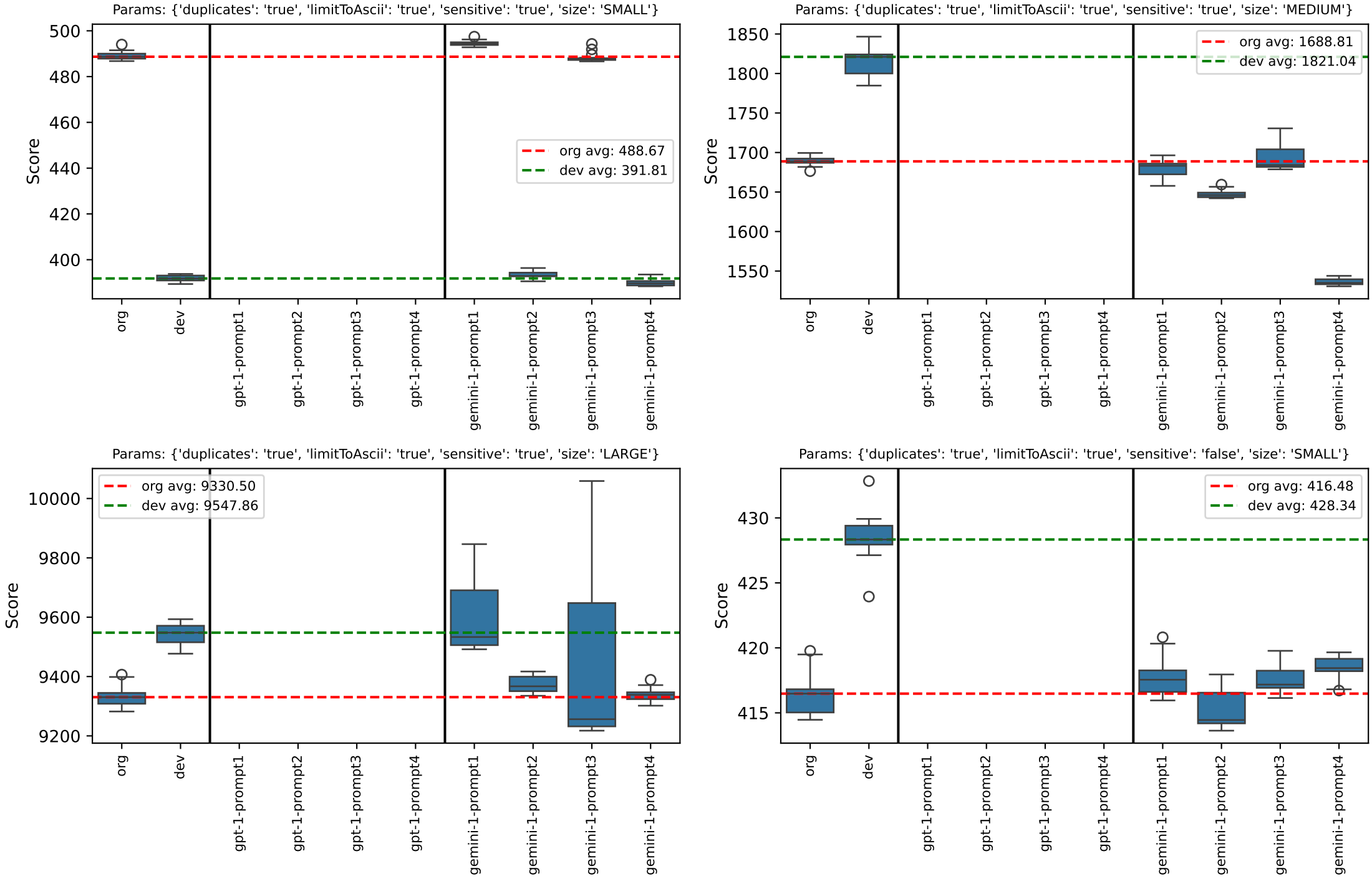
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.shortAsciiFirst



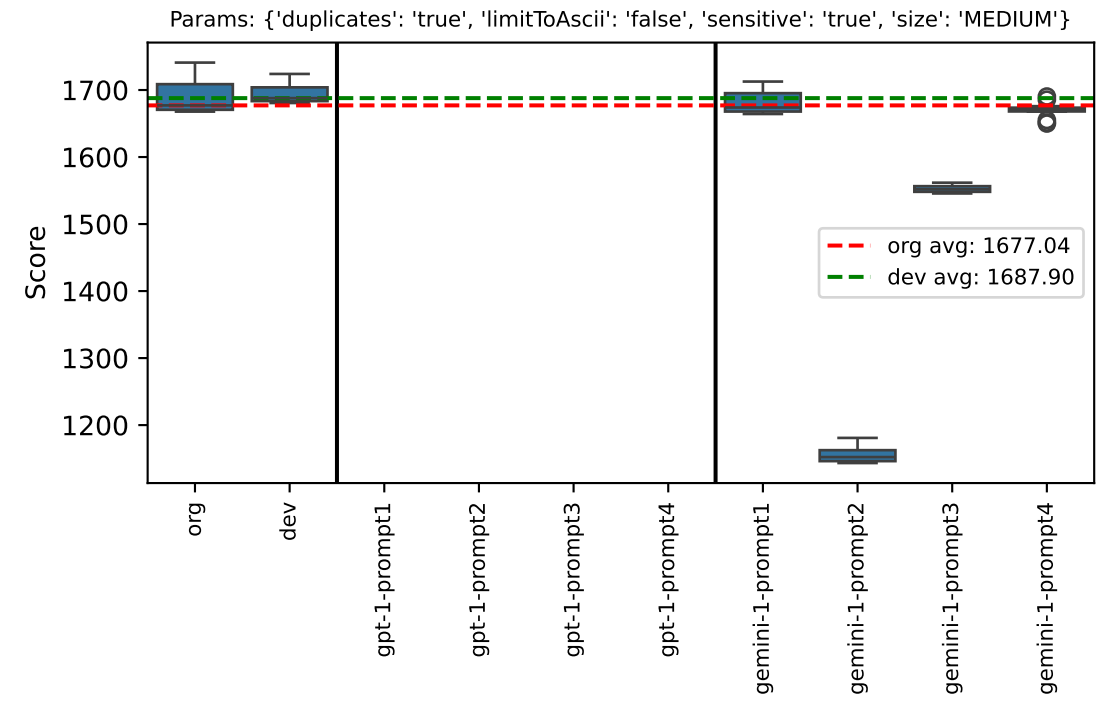
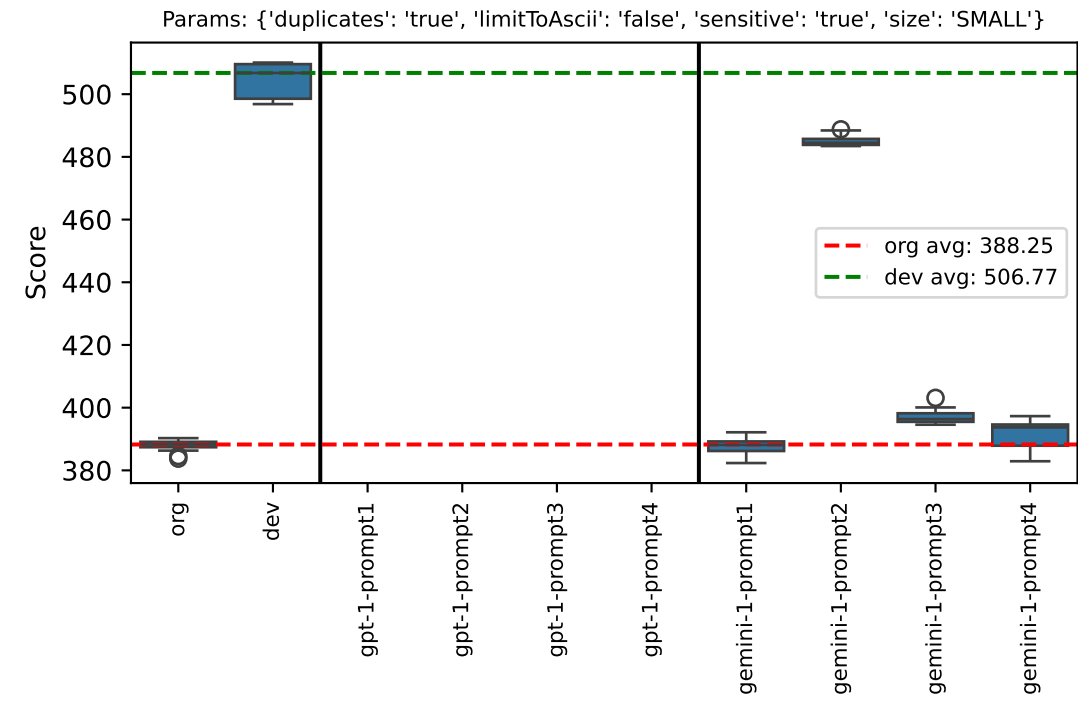
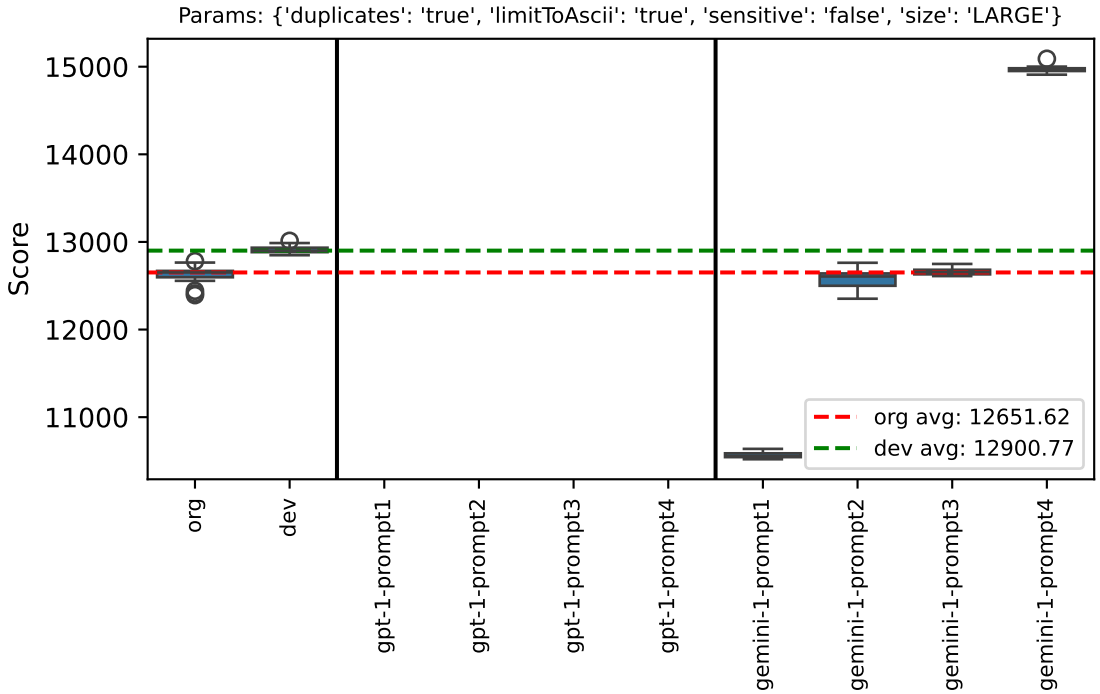
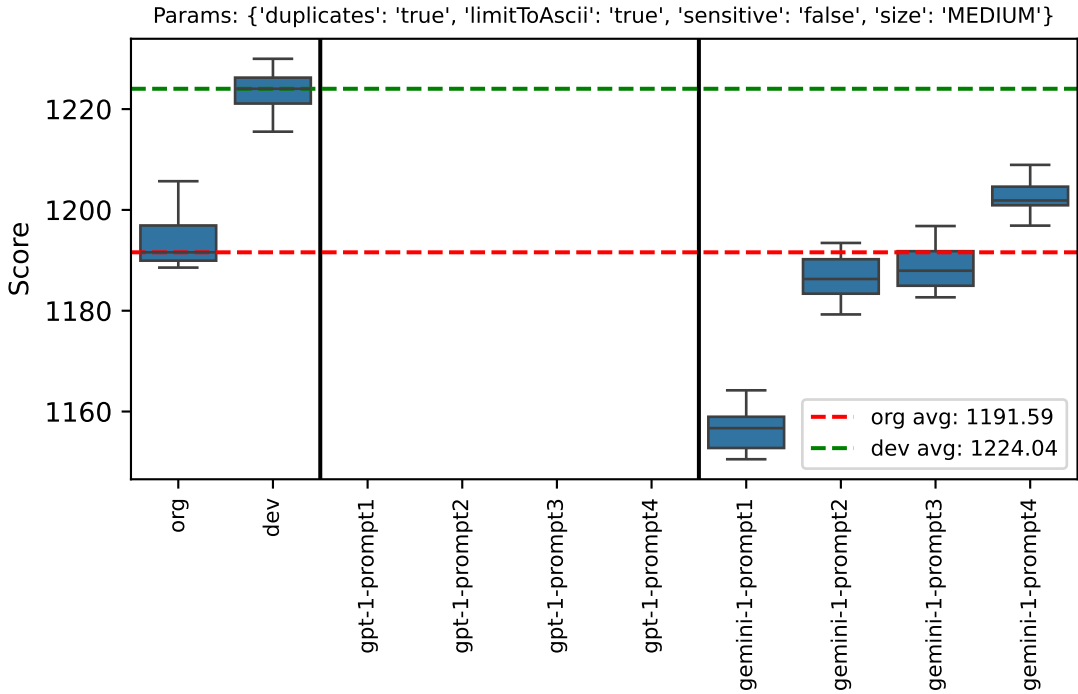
Benchmark: io.netty.handler.codec.http.QueryStringEncoderBenchmark.shortUtf8



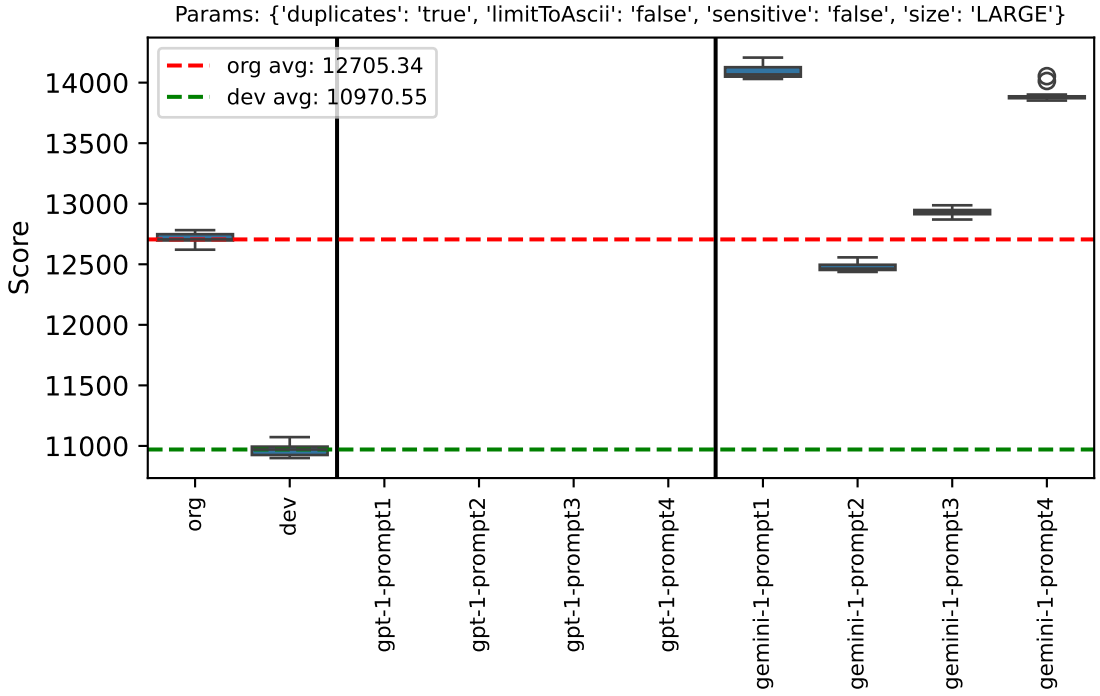
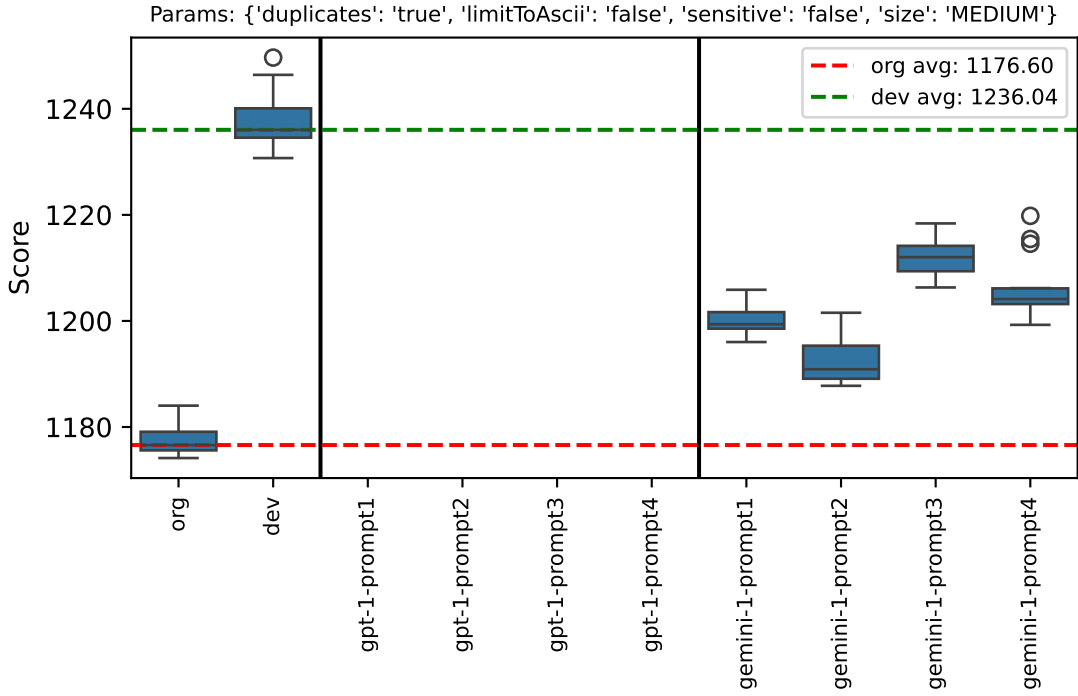
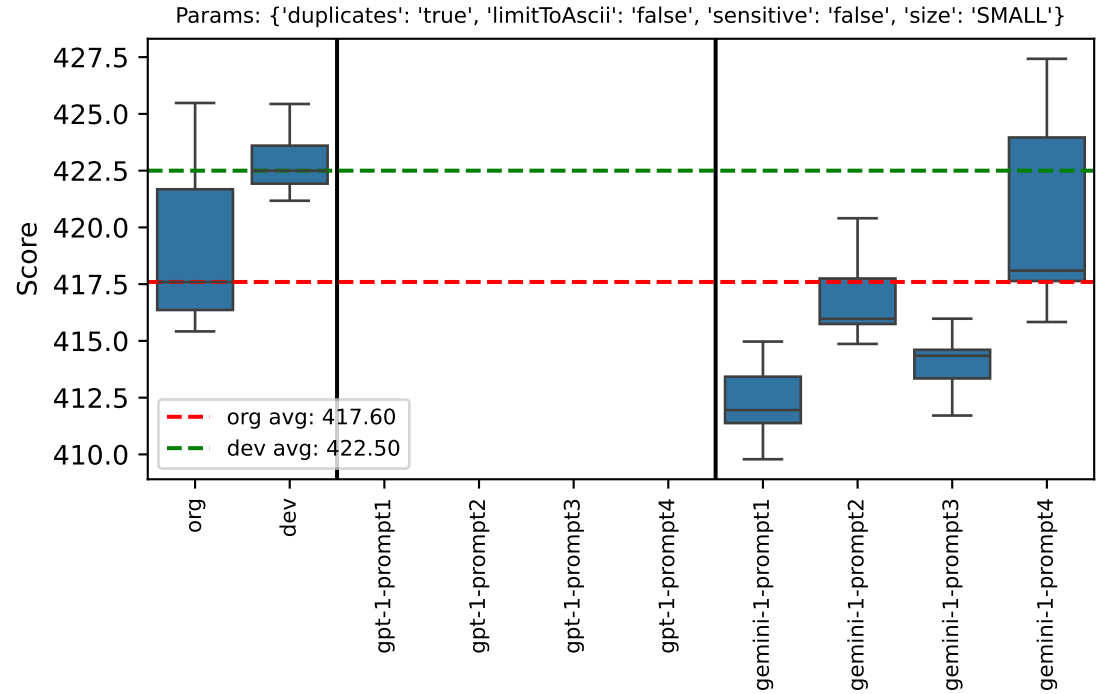
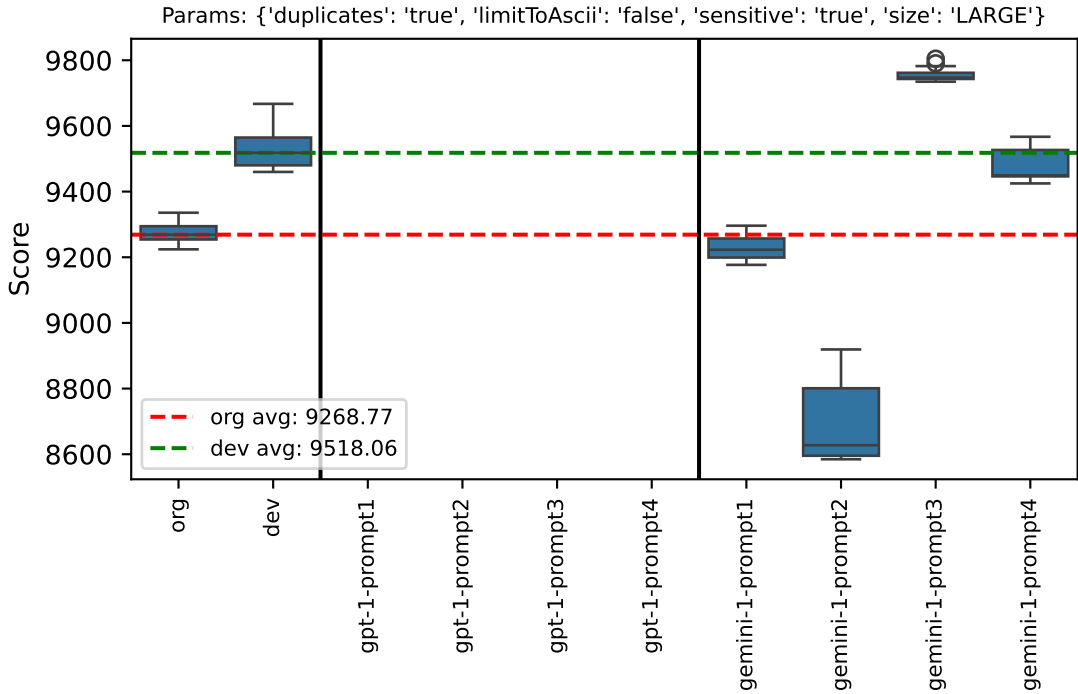
Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode



Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode

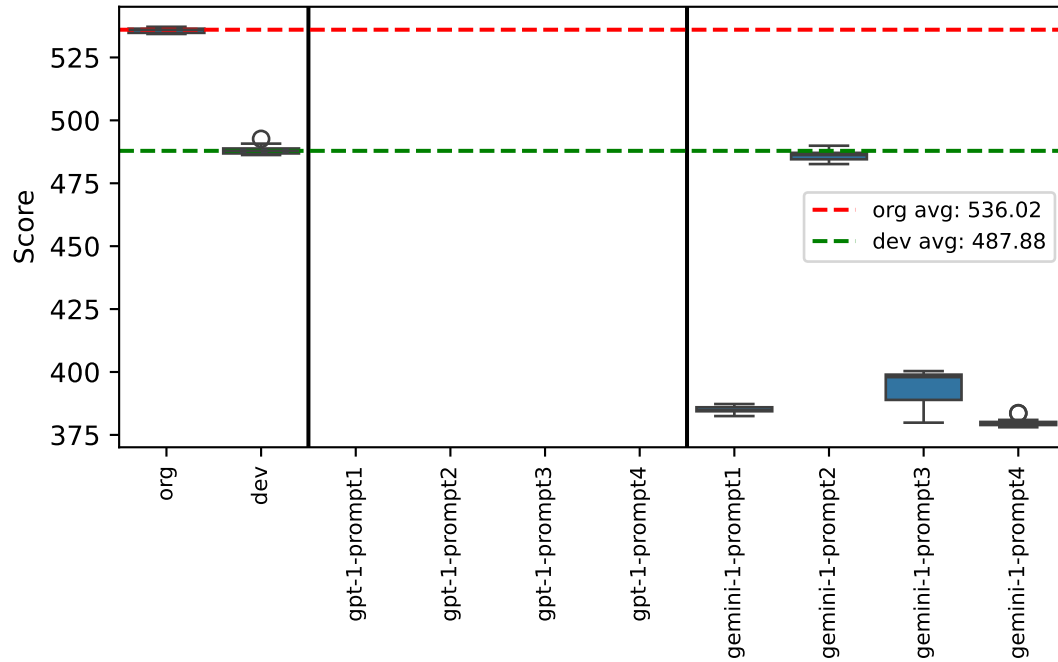


Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode

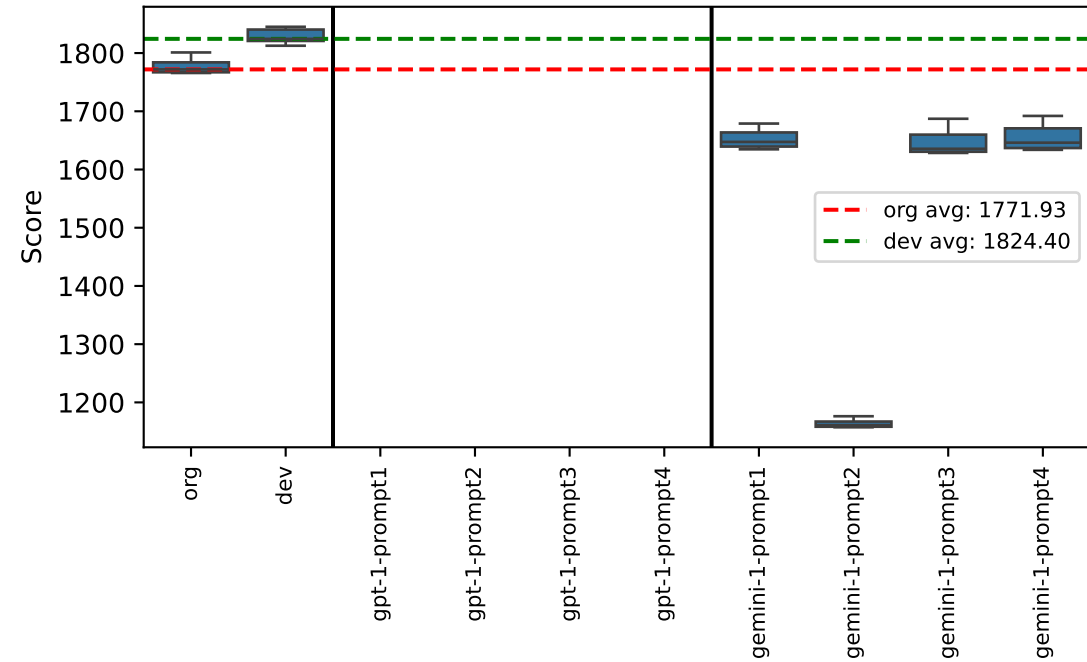


Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode

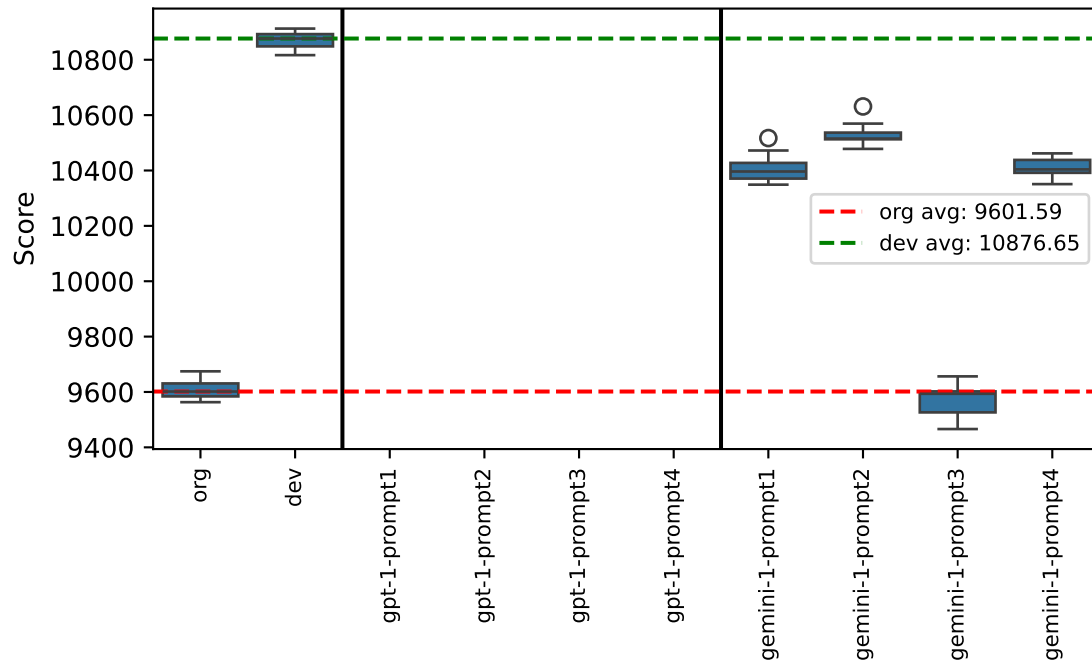
Params: {'duplicates': 'false', 'limitToAscii': 'true', 'sensitive': 'true', 'size': 'SMALL'}



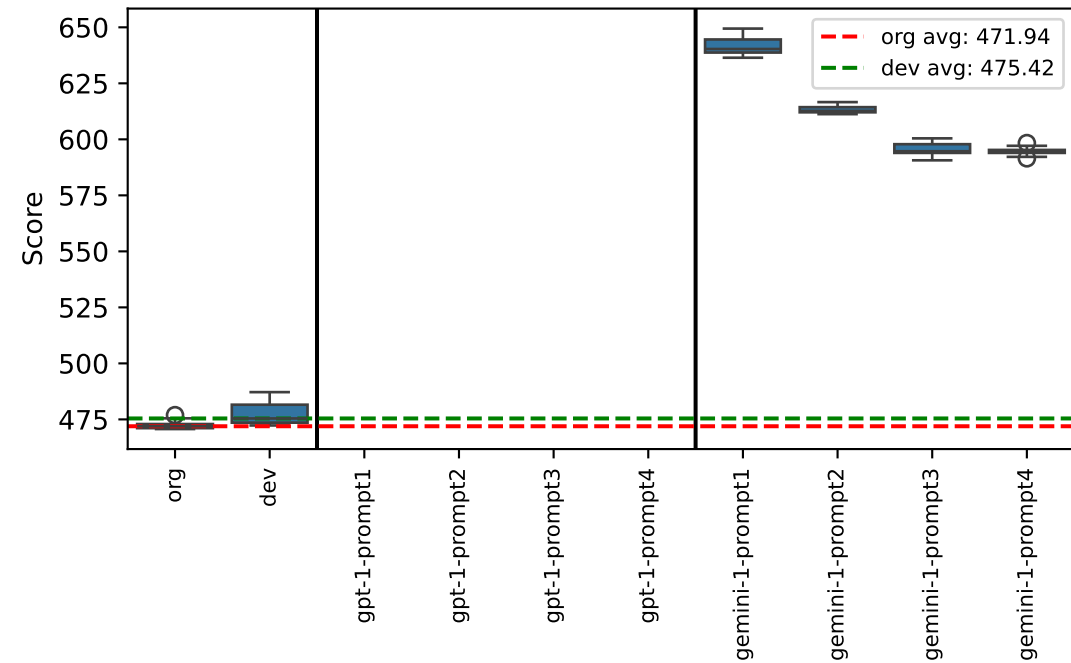
Params: {'duplicates': 'false', 'limitToAscii': 'true', 'sensitive': 'true', 'size': 'MEDIUM'}



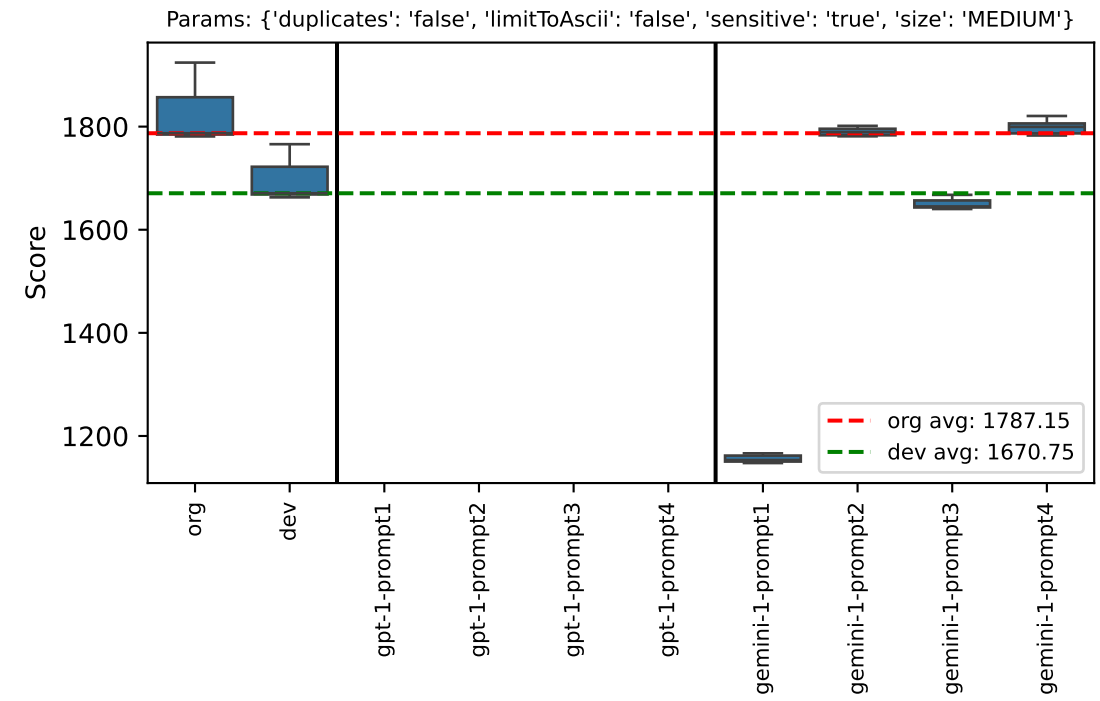
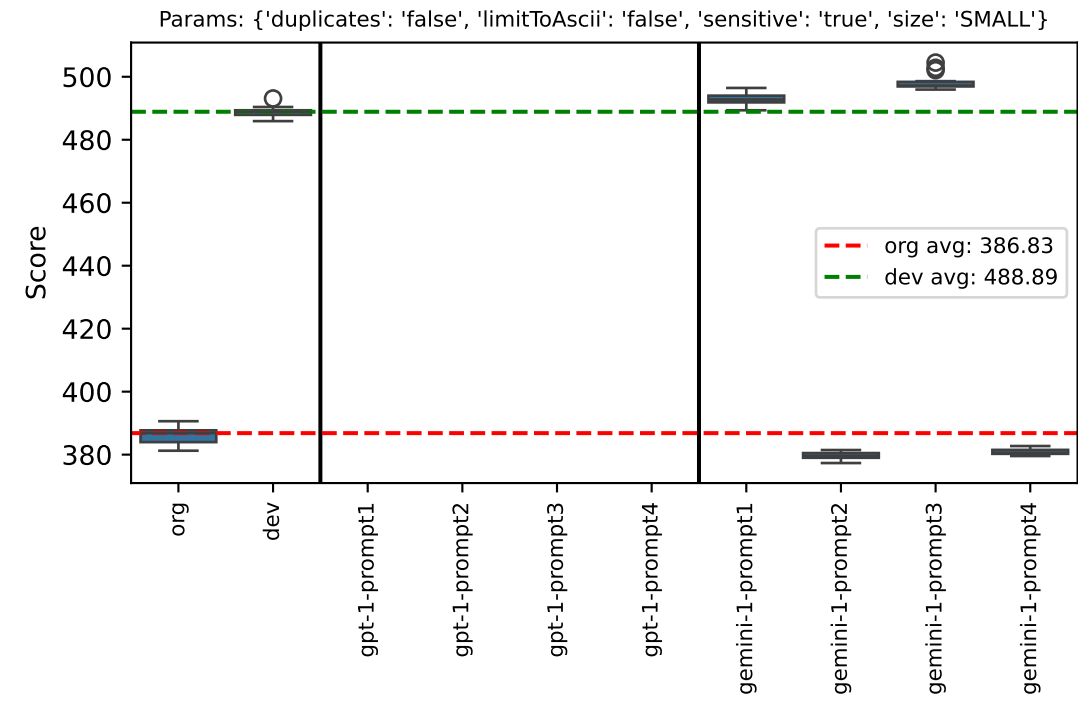
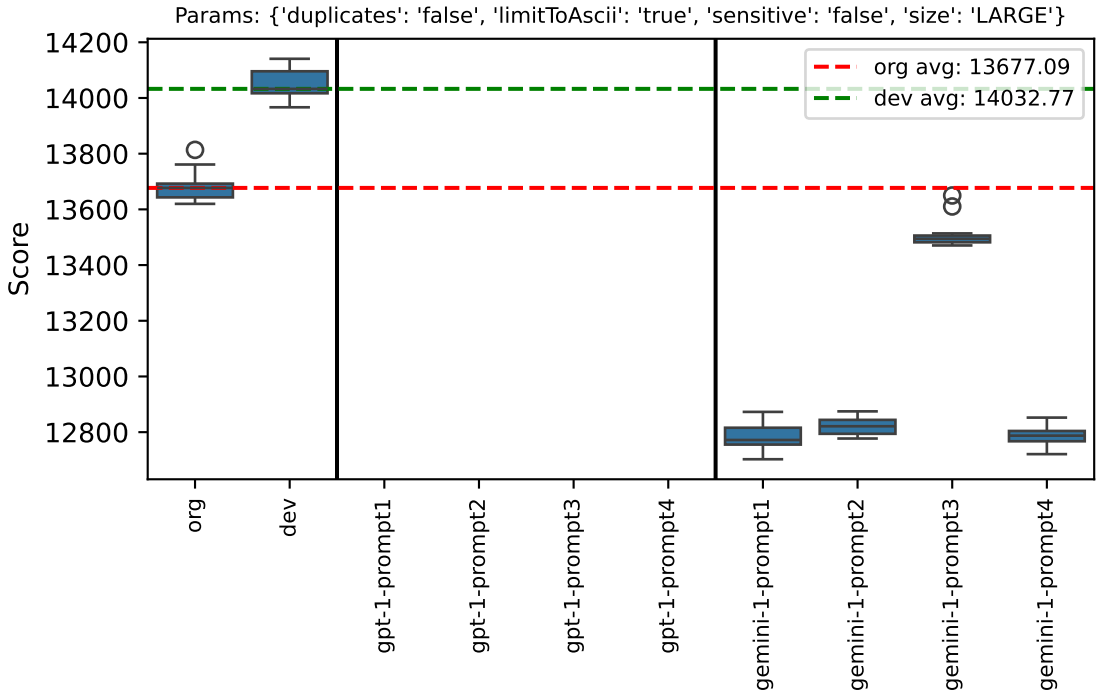
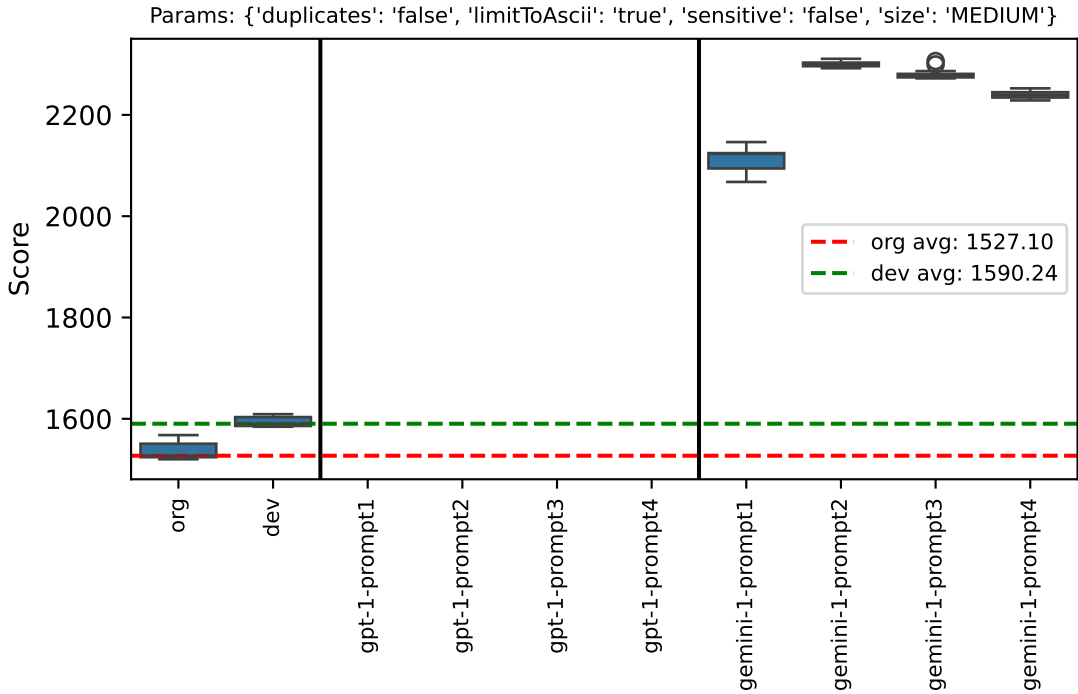
Params: {'duplicates': 'false', 'limitToAscii': 'true', 'sensitive': 'true', 'size': 'LARGE'}



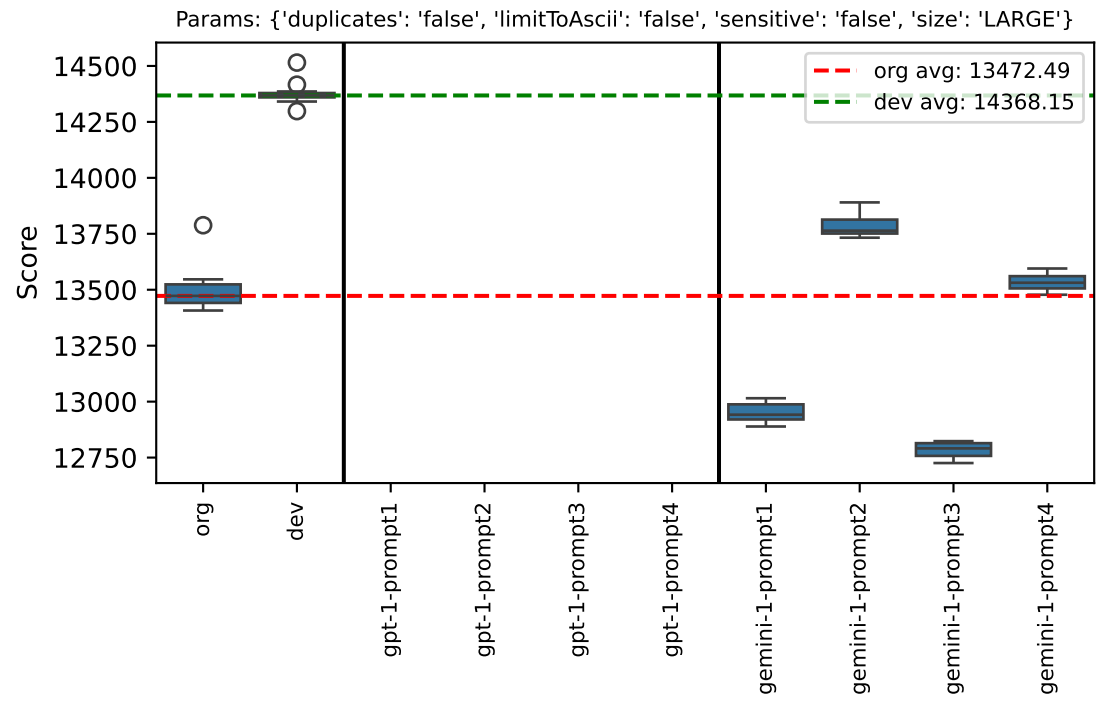
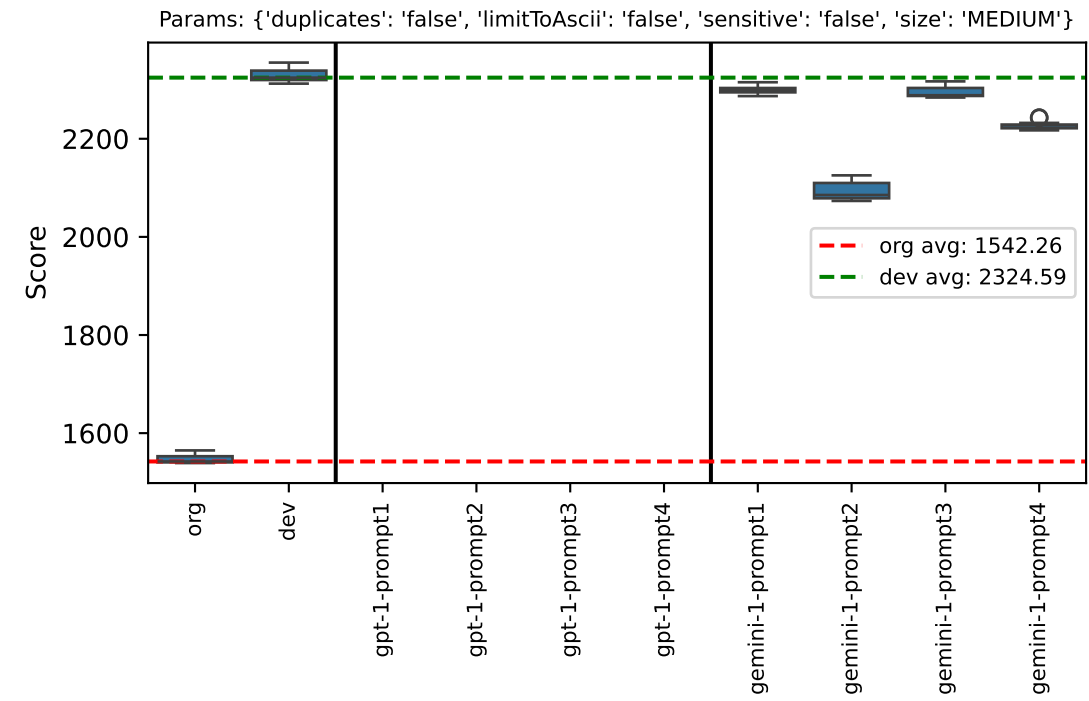
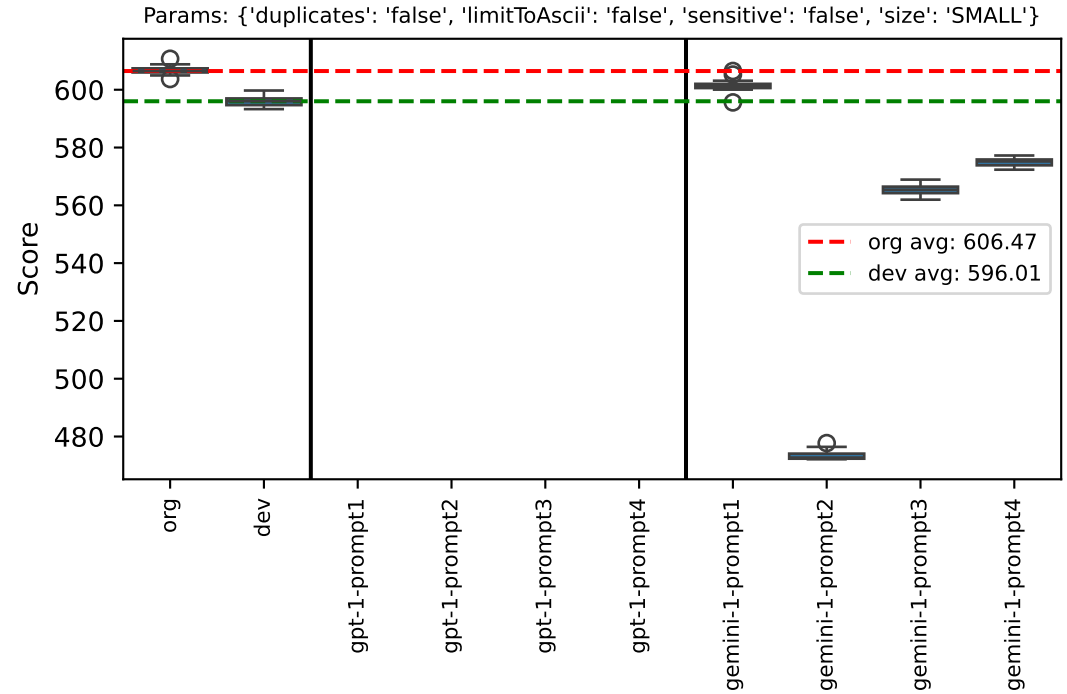
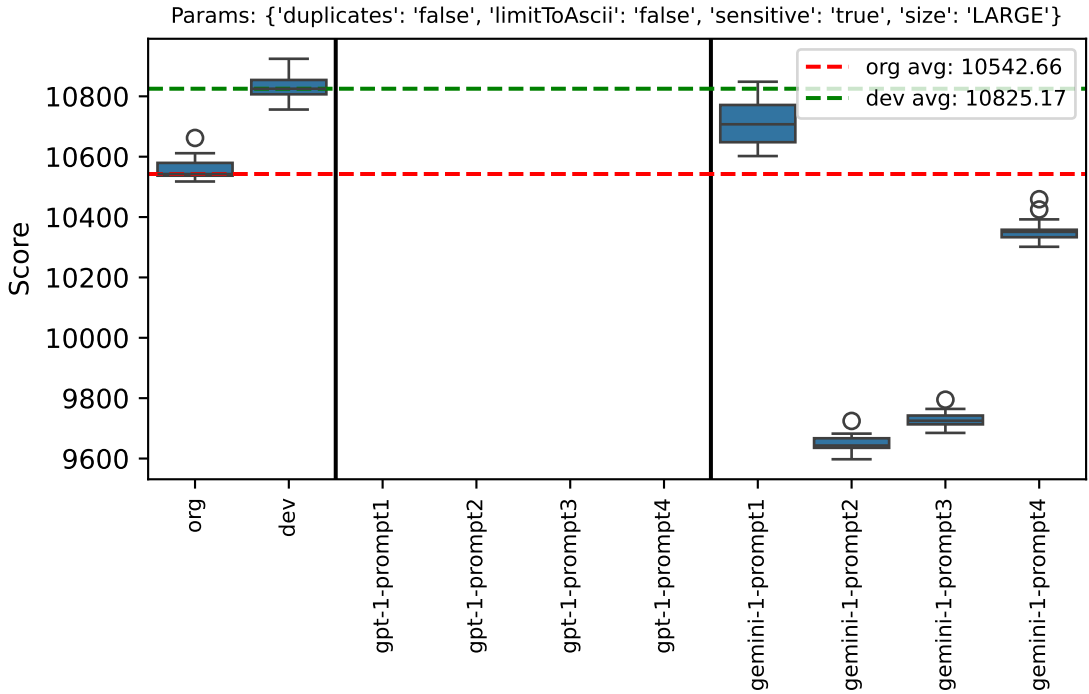
Params: {'duplicates': 'false', 'limitToAscii': 'true', 'sensitive': 'false', 'size': 'SMALL'}



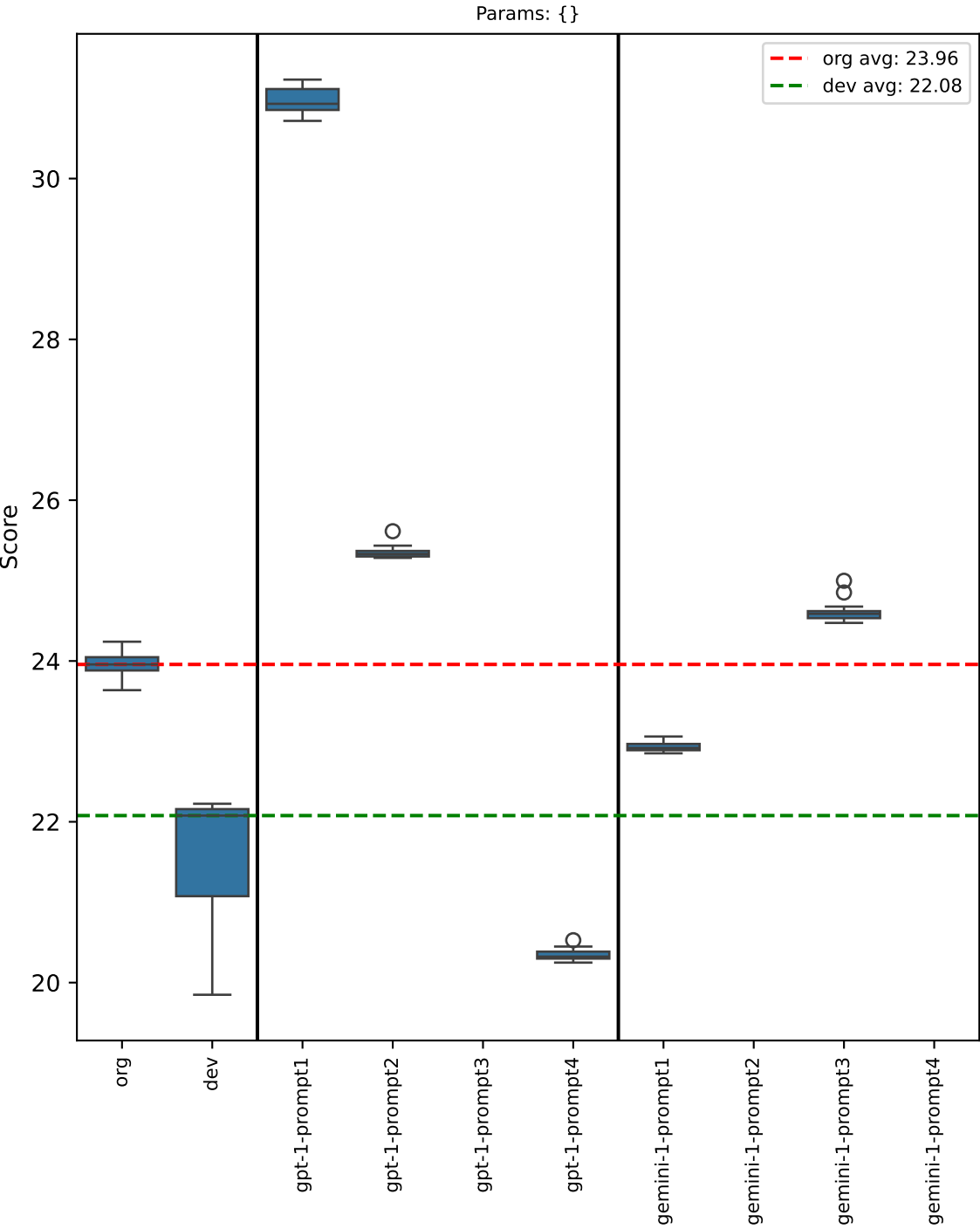
Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode



Benchmark: io.netty.handler.codec.http2.HpackEncoderBenchmark.encode

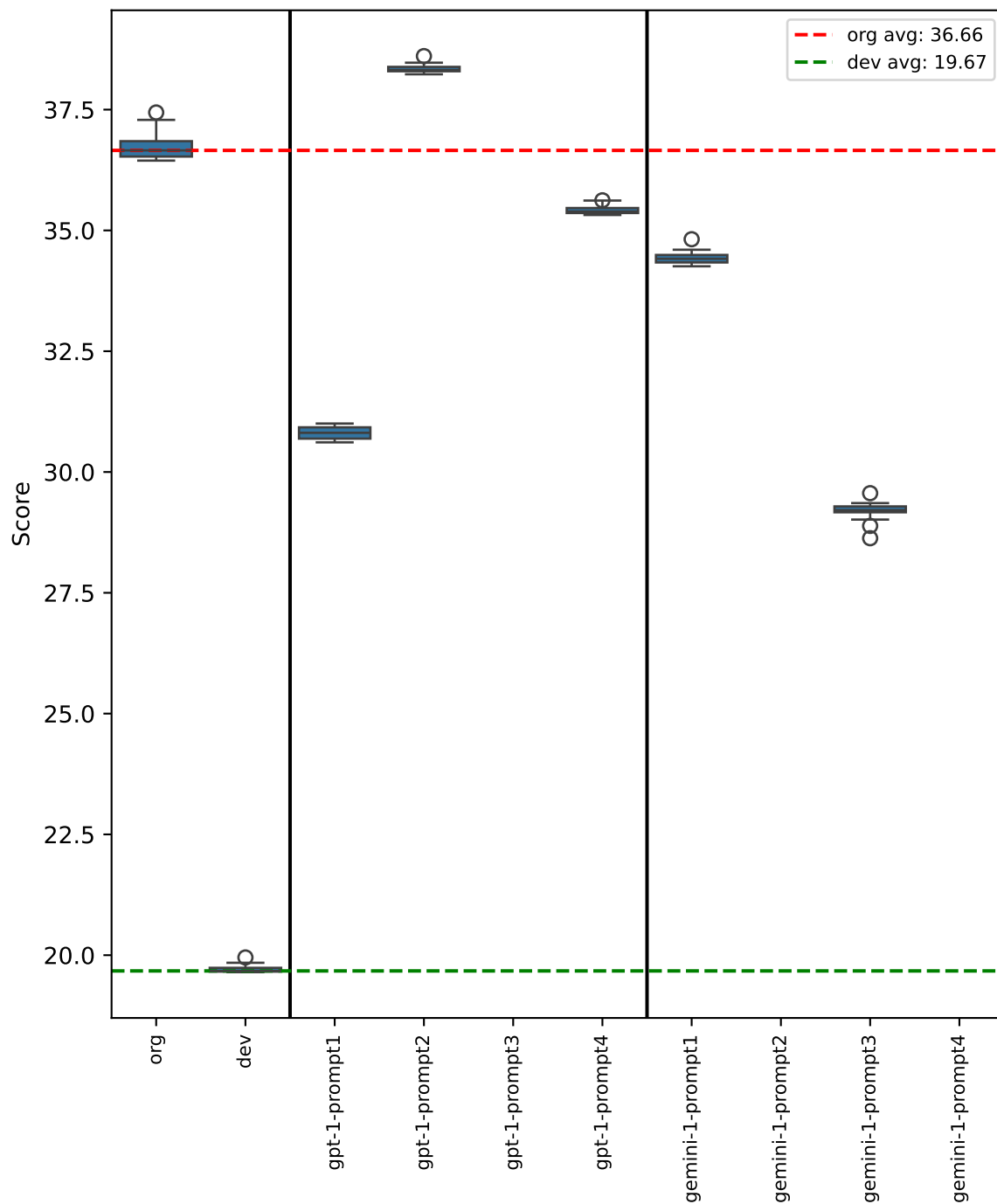


Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupHttp

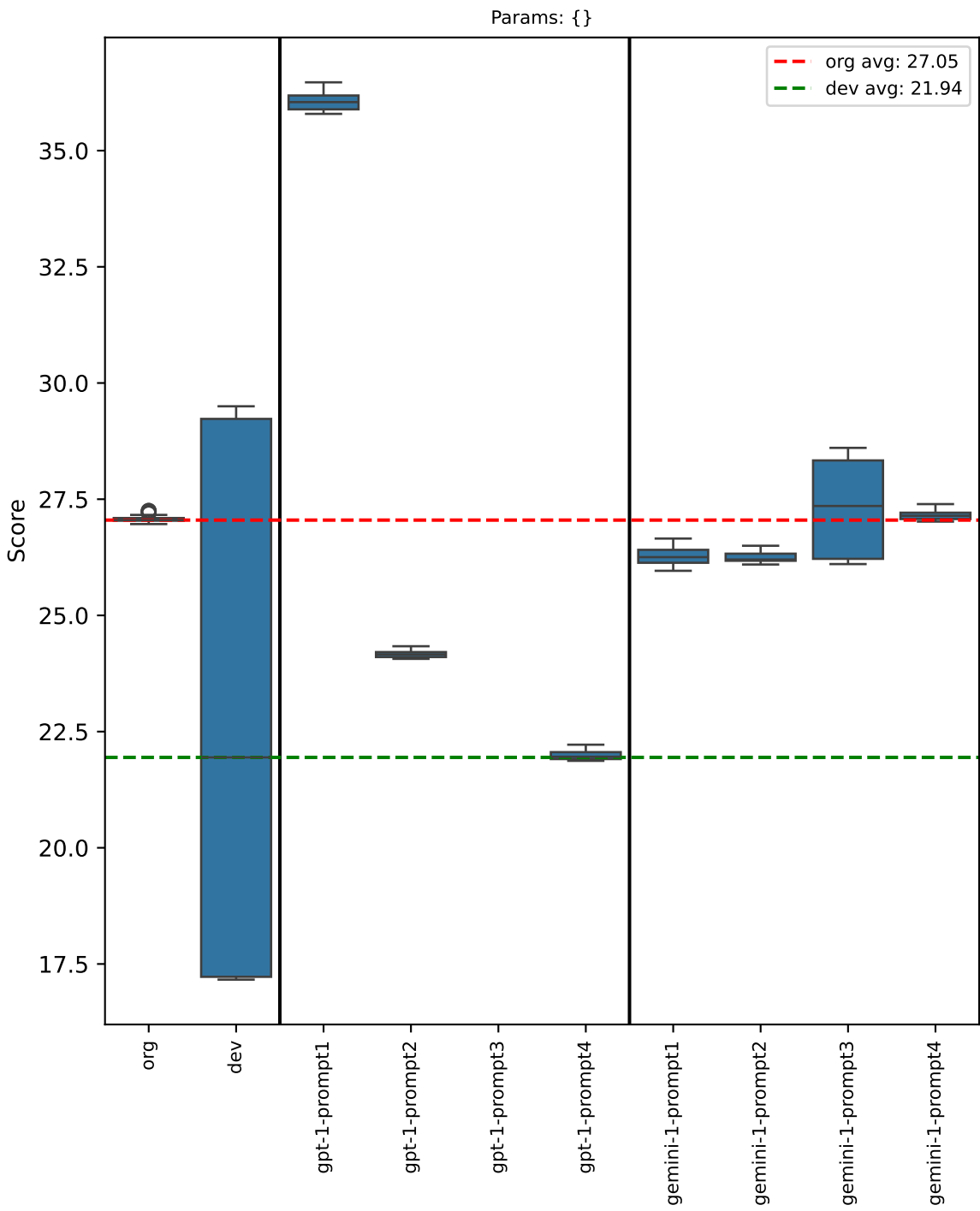


Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupHttps

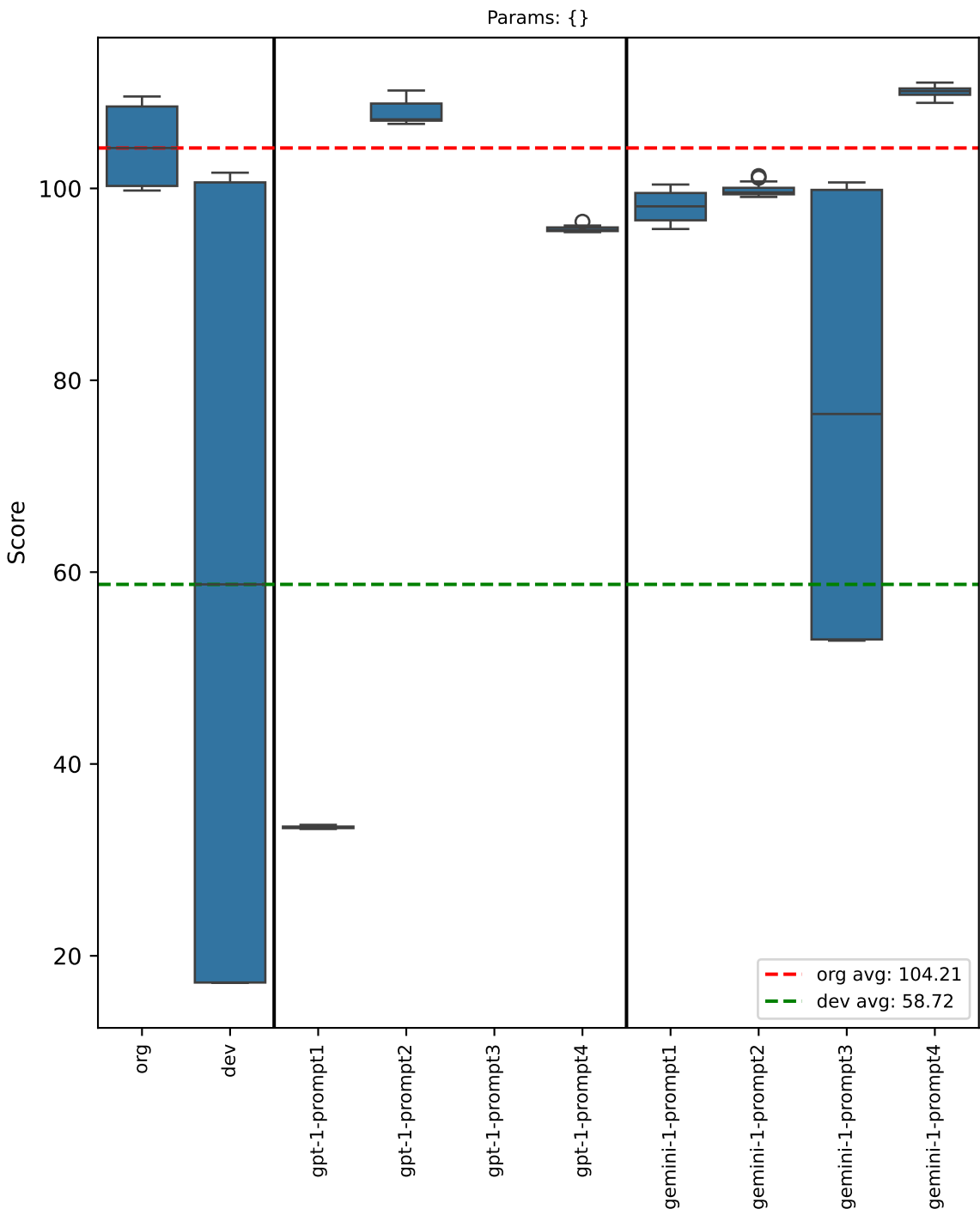
Params: {}



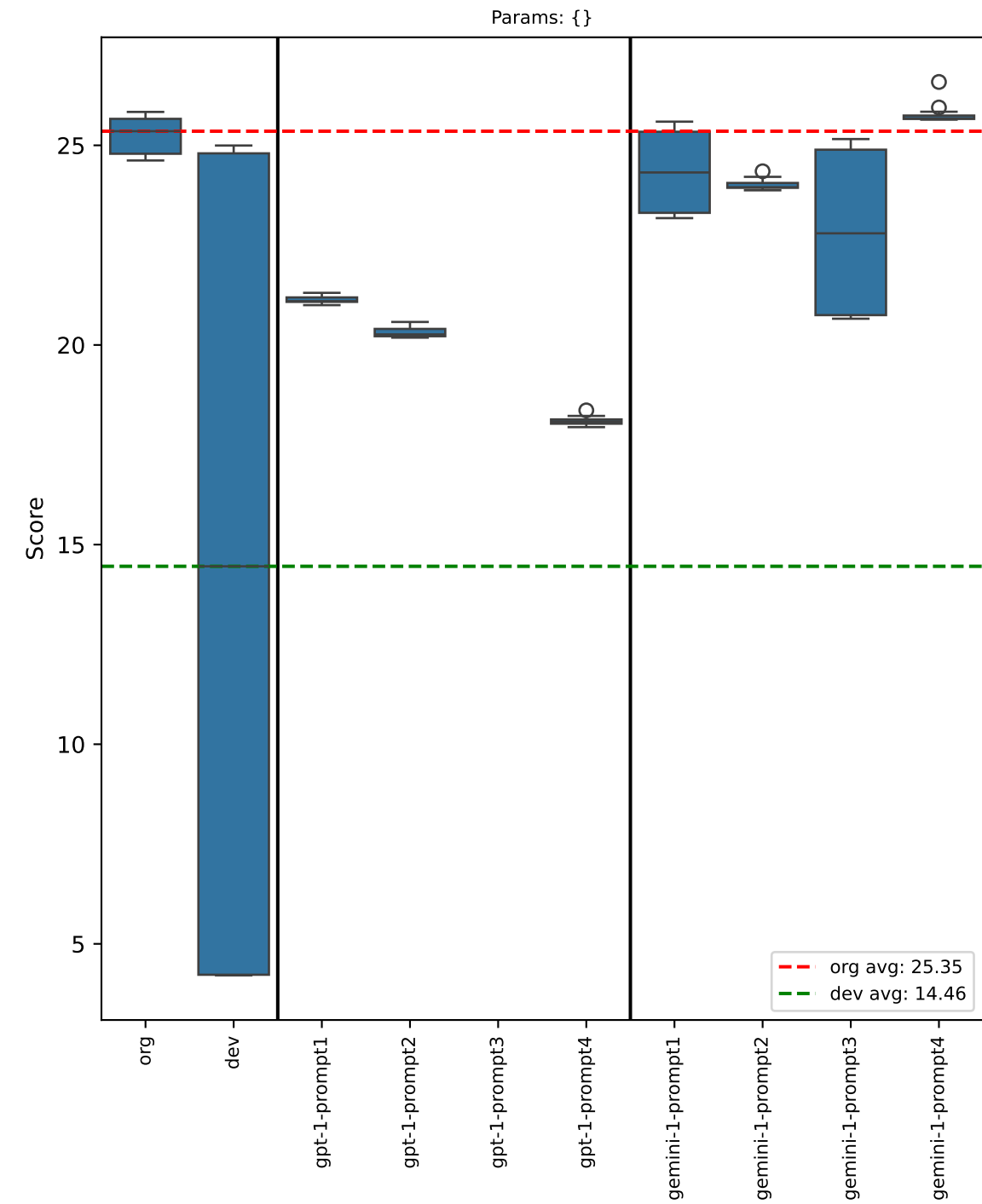
Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupNameAndValueMatchFirst



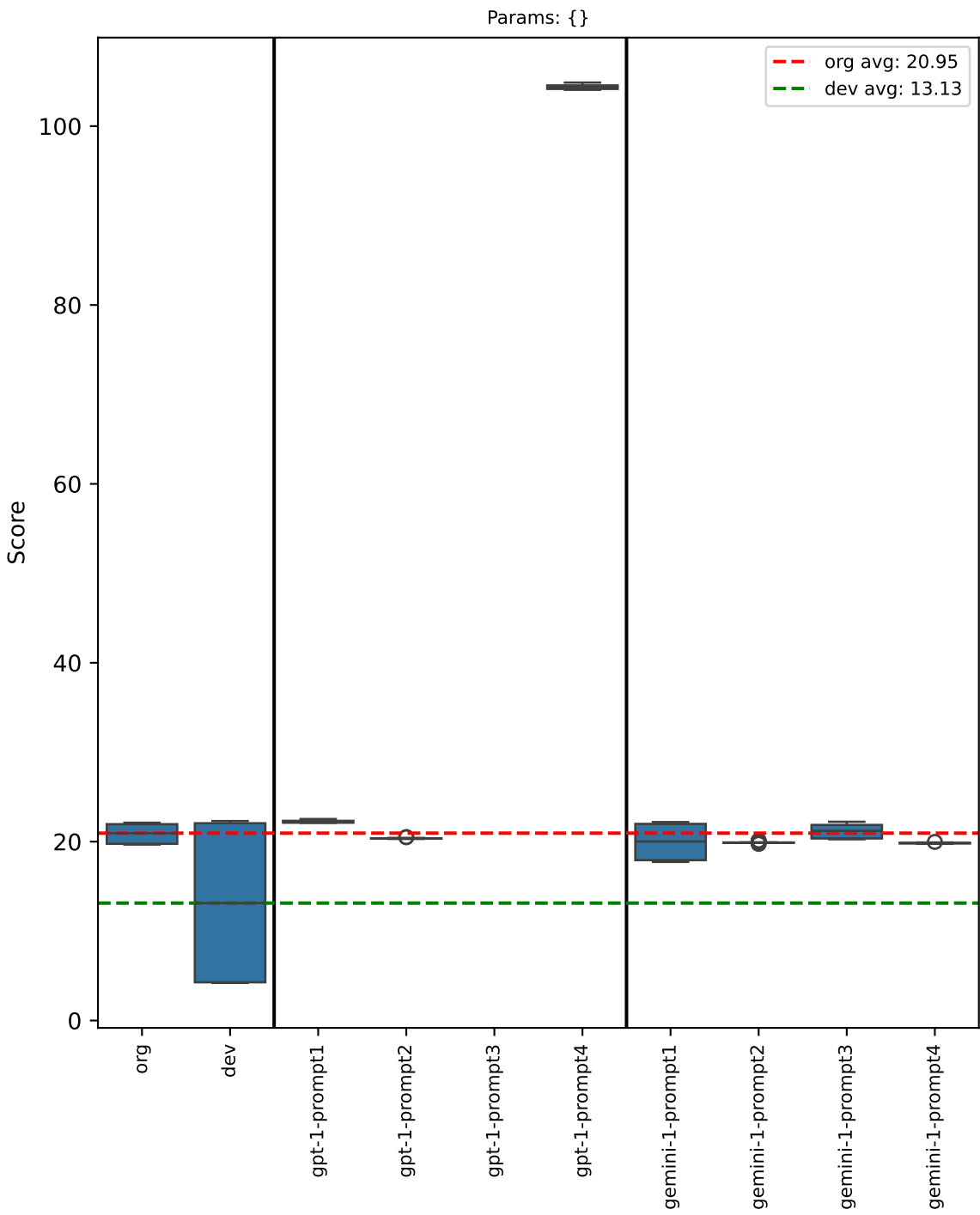
Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupNameAndValueMatchLast



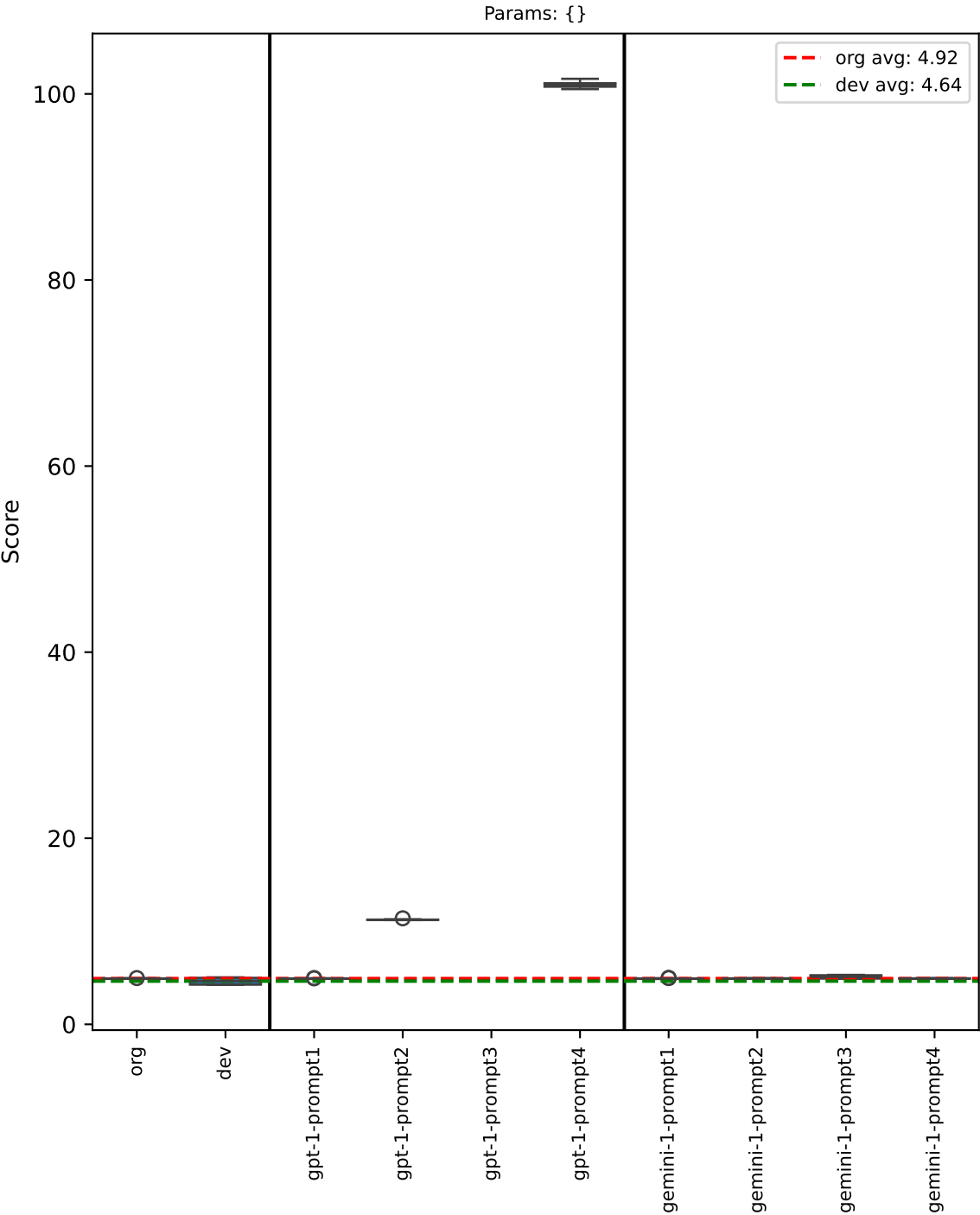
Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupNameOnlyMatchBeginTable



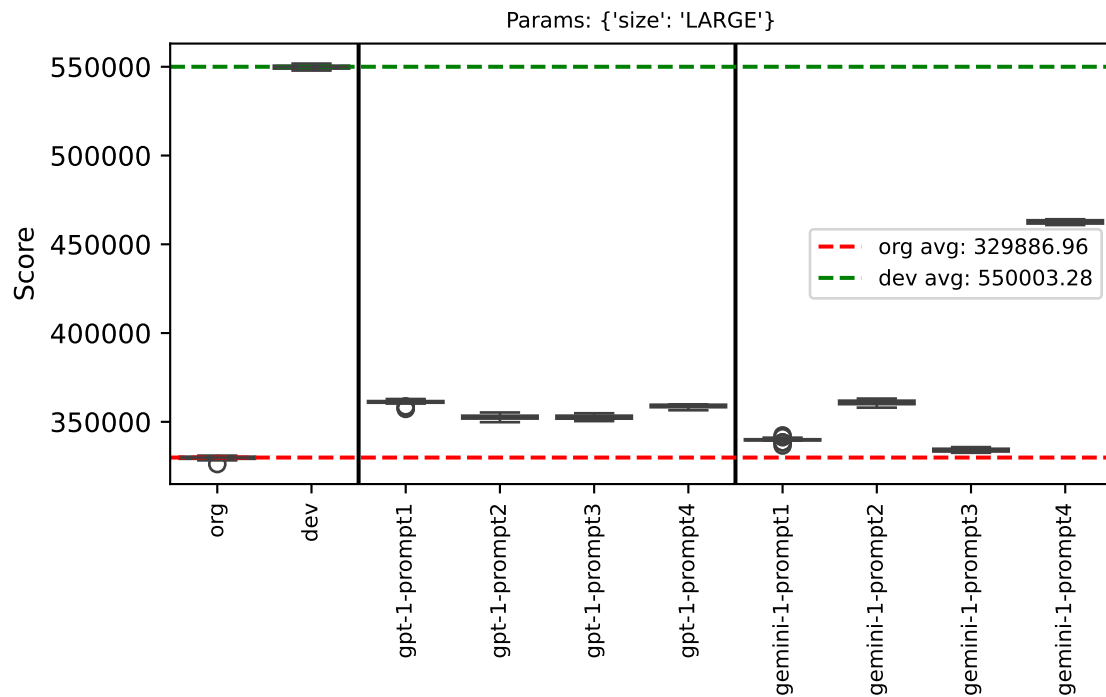
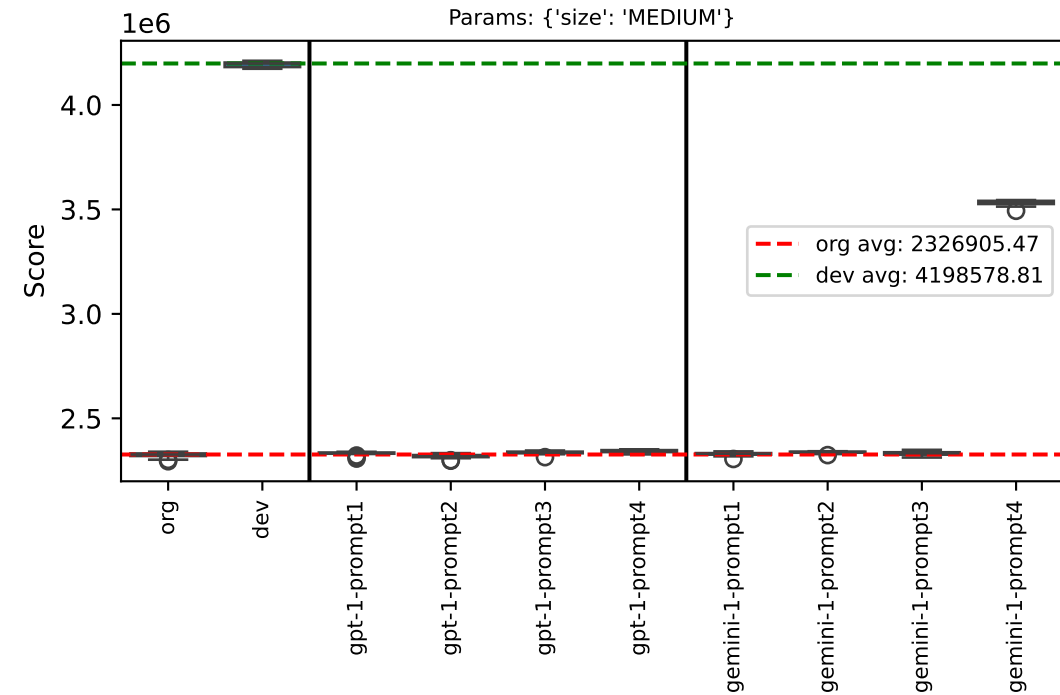
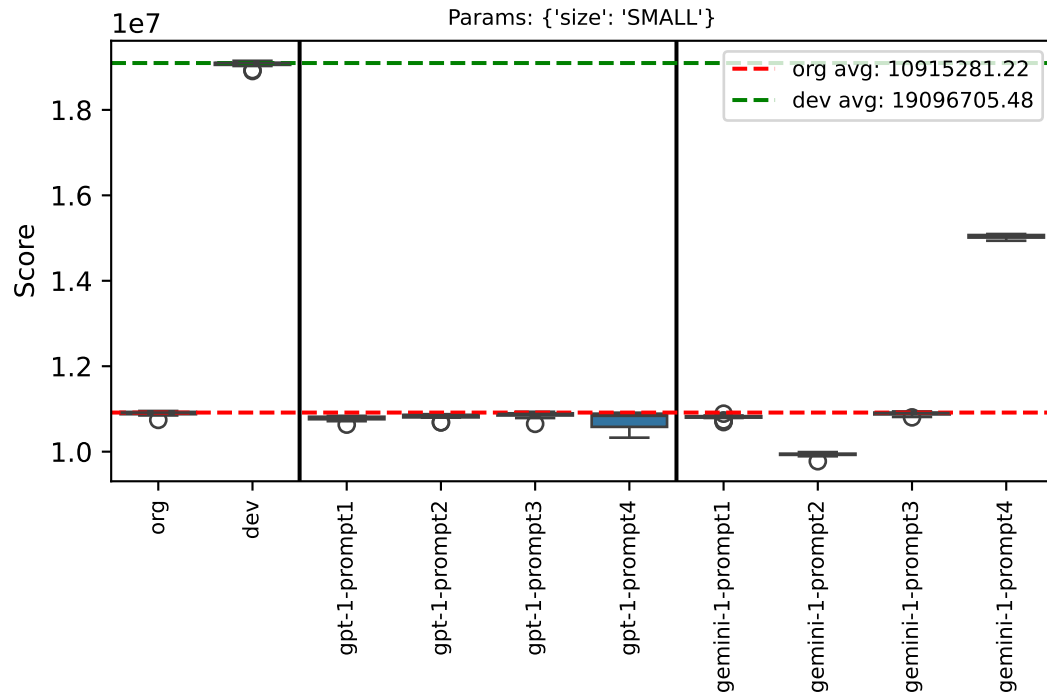
Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupNameOnlyMatchEndTable



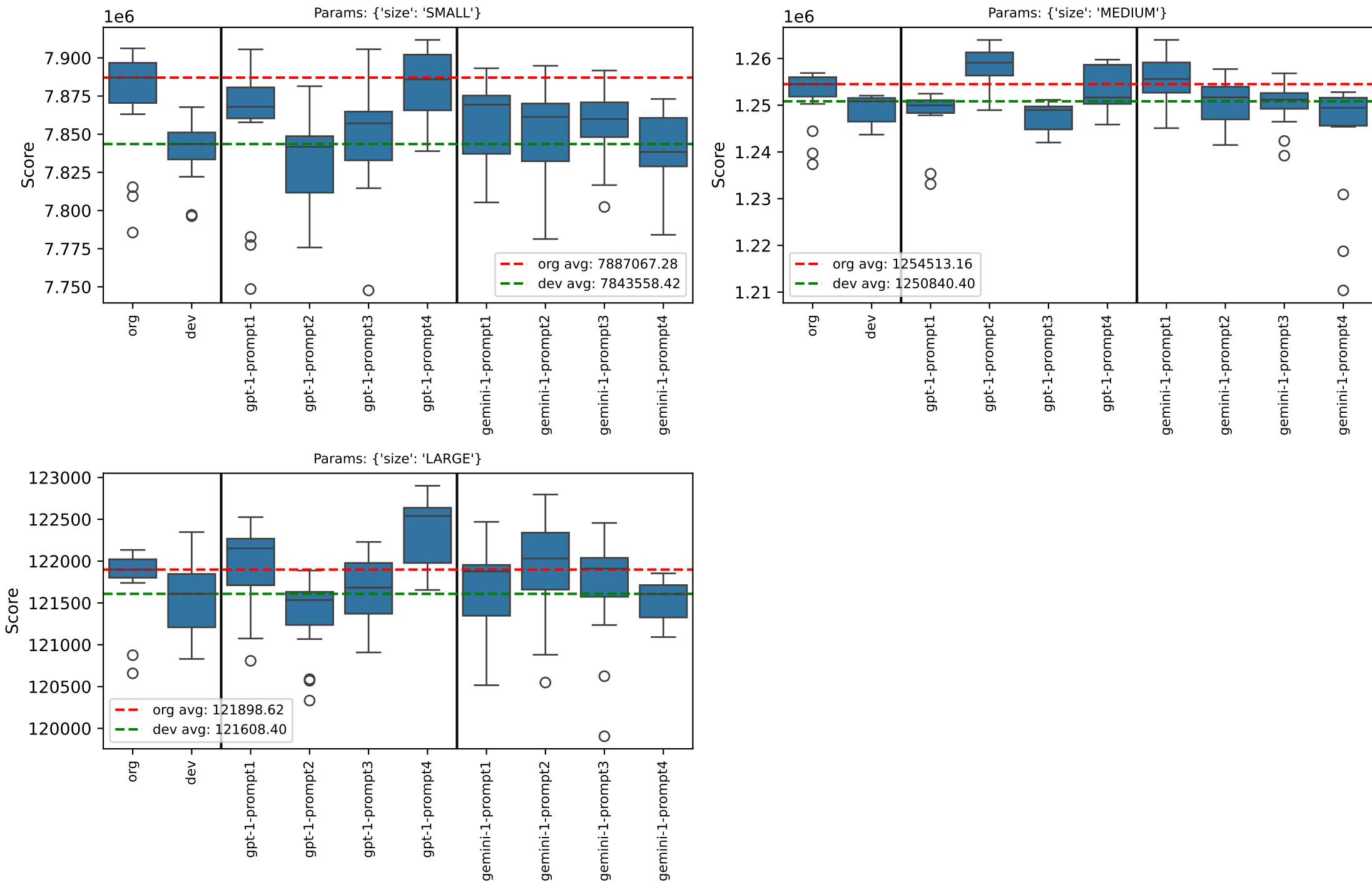
Benchmark: io.netty.handler.codec.http2.HpackStaticTableBenchmark.lookupNoNameMatch



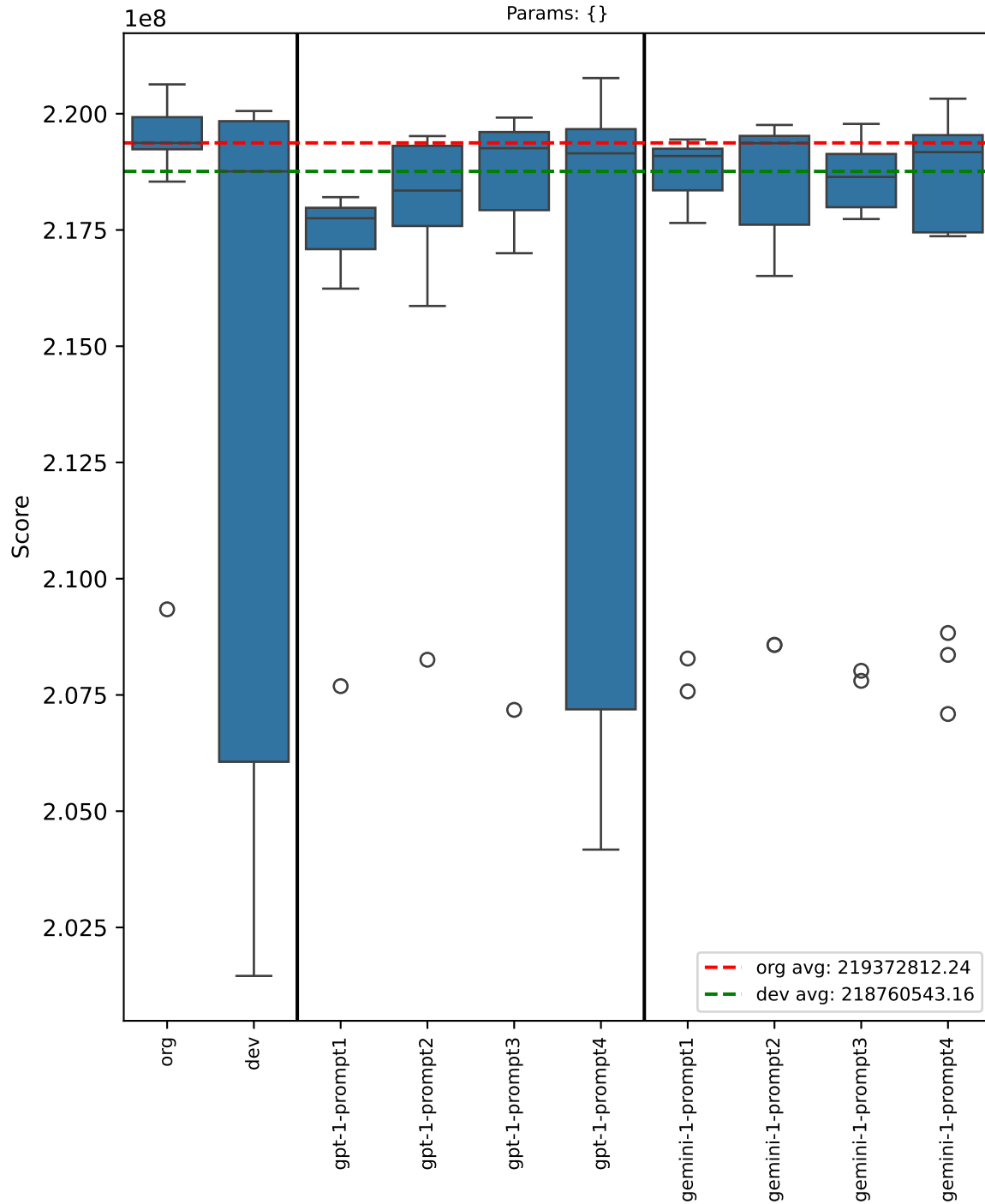
Benchmark: io.netty.handler.codec.http2.HpackUtilBenchmark.newEquals



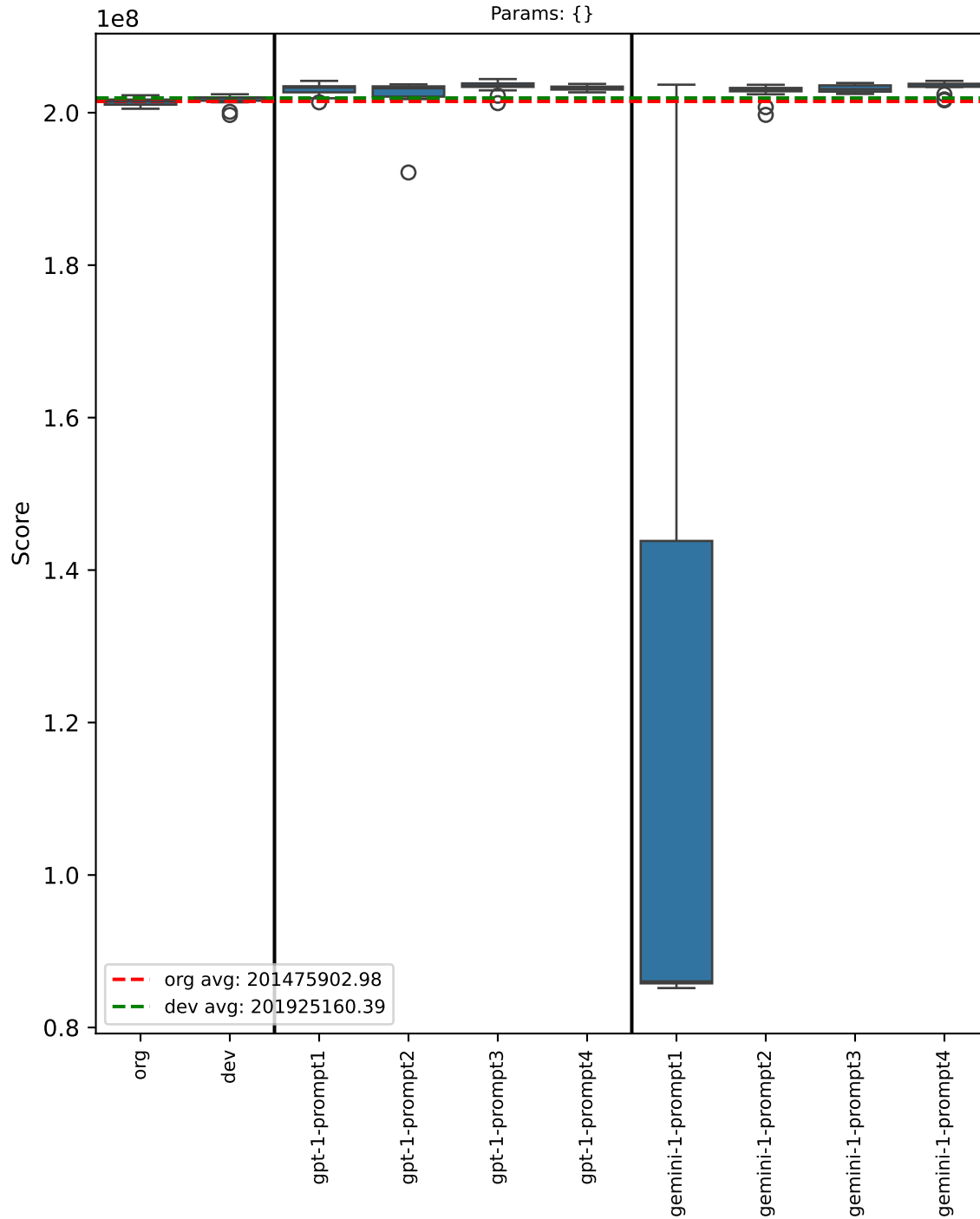
Benchmark: io.netty.handler.codec.http2.HpackUtilBenchmark.oldEquals



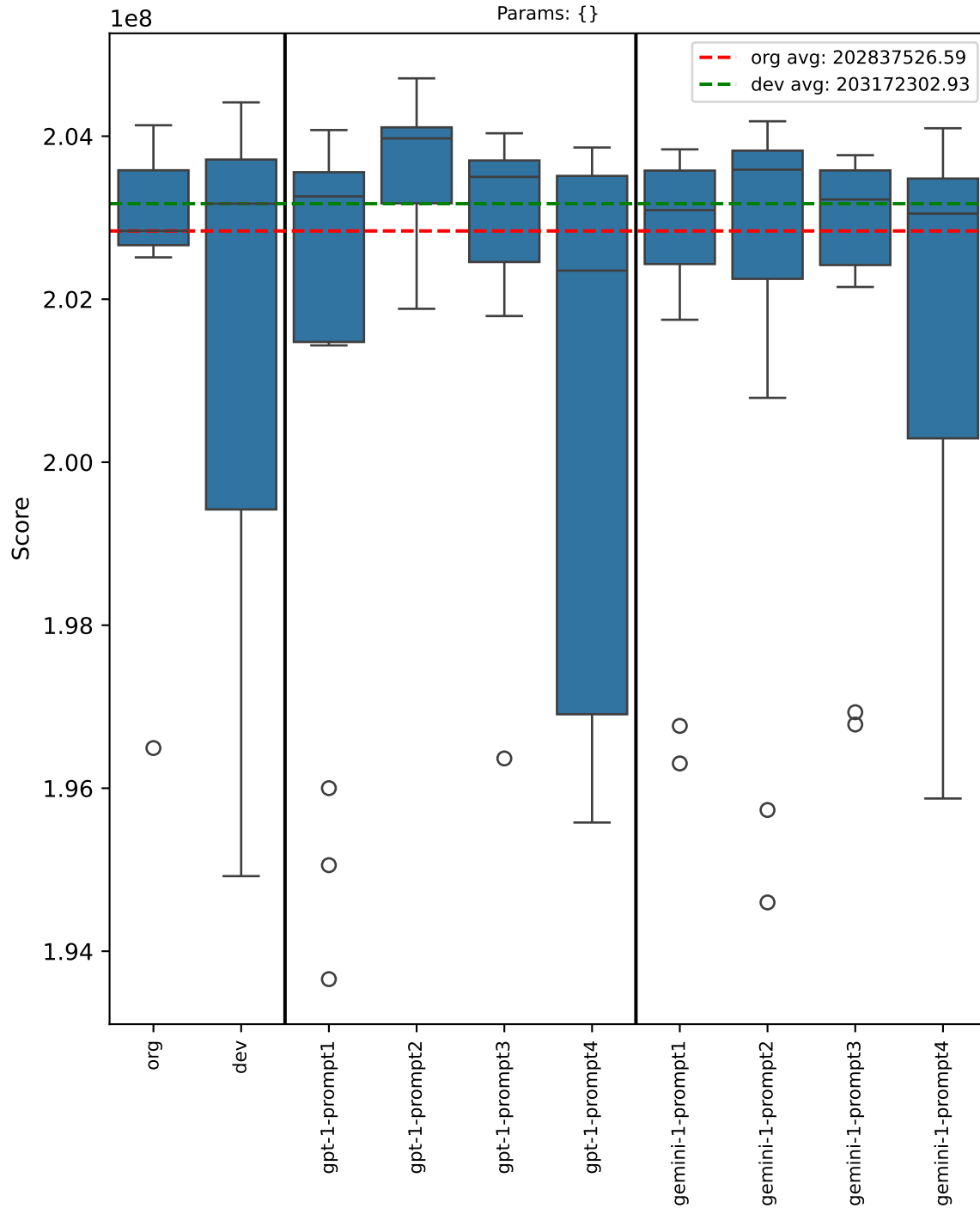
Benchmark: io.netty.microbench.buffer.UnsafeByteBufferBenchmark.setGetLongByteBuffer



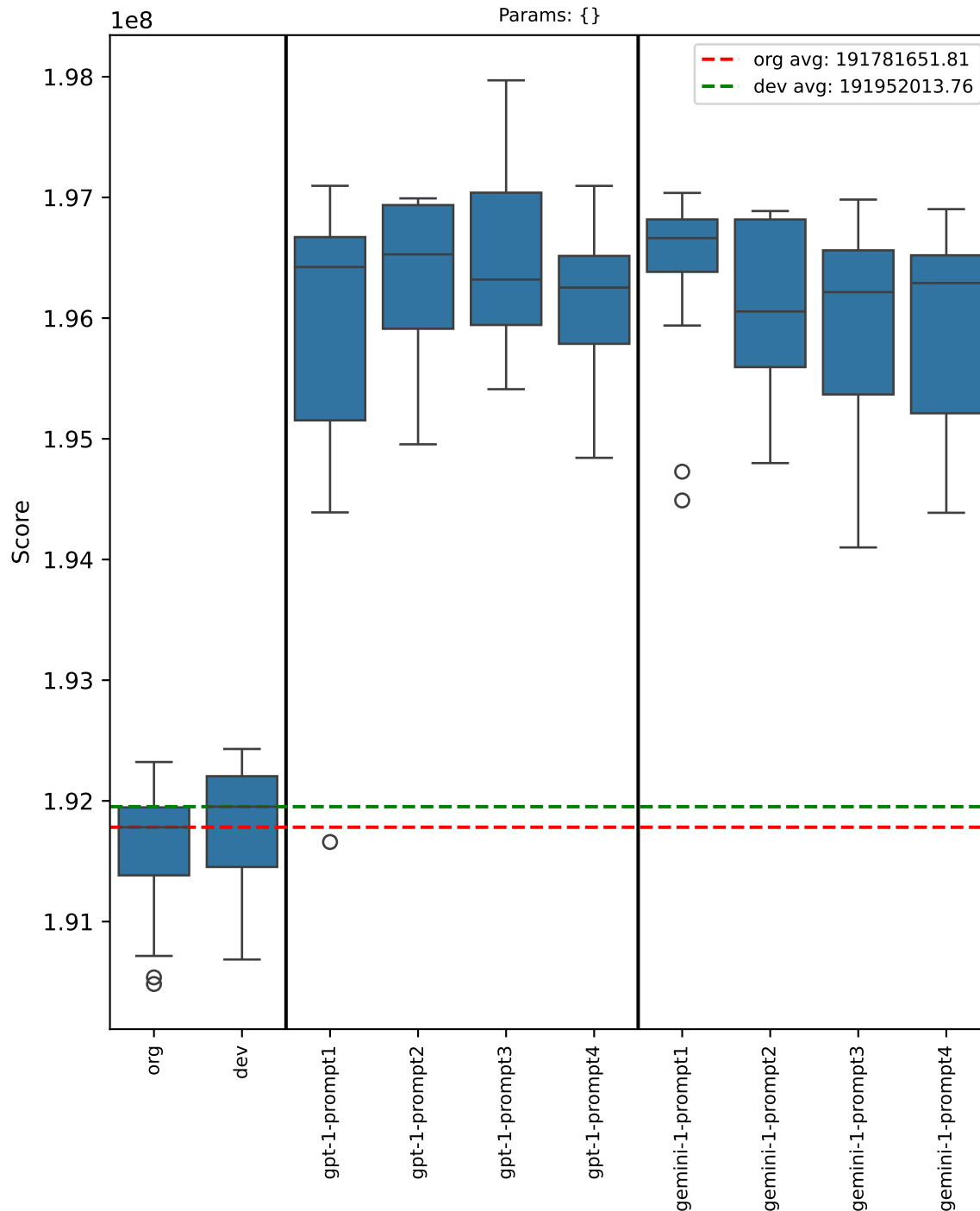
Benchmark: io.netty.microbench.buffer.UnsafeByteBufferBenchmark.setGetLongUnsafeByteBuffer



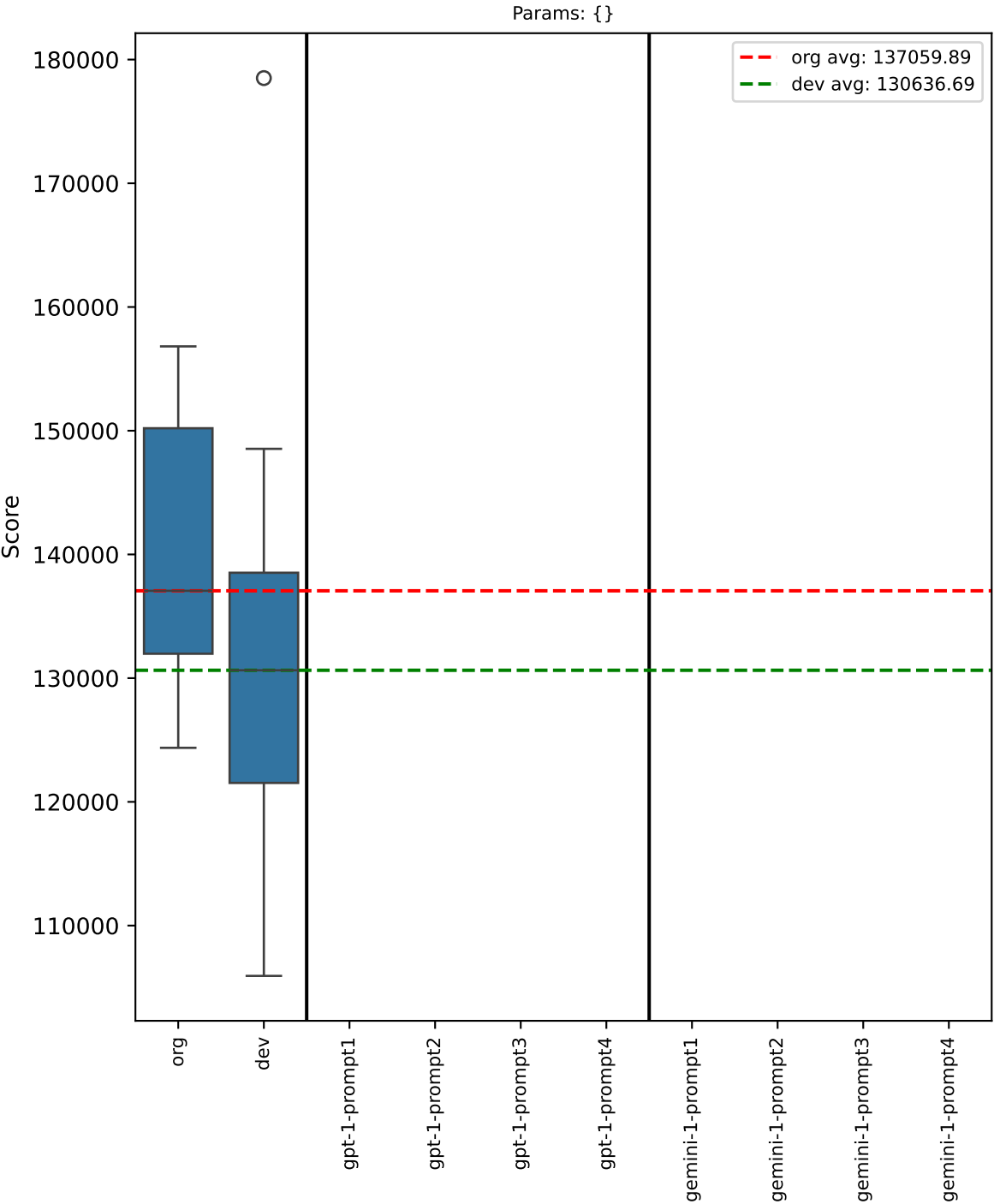
Benchmark: io.netty.microbench.buffer.UnsafeByteBufferBenchmark.setLongByteBuffer



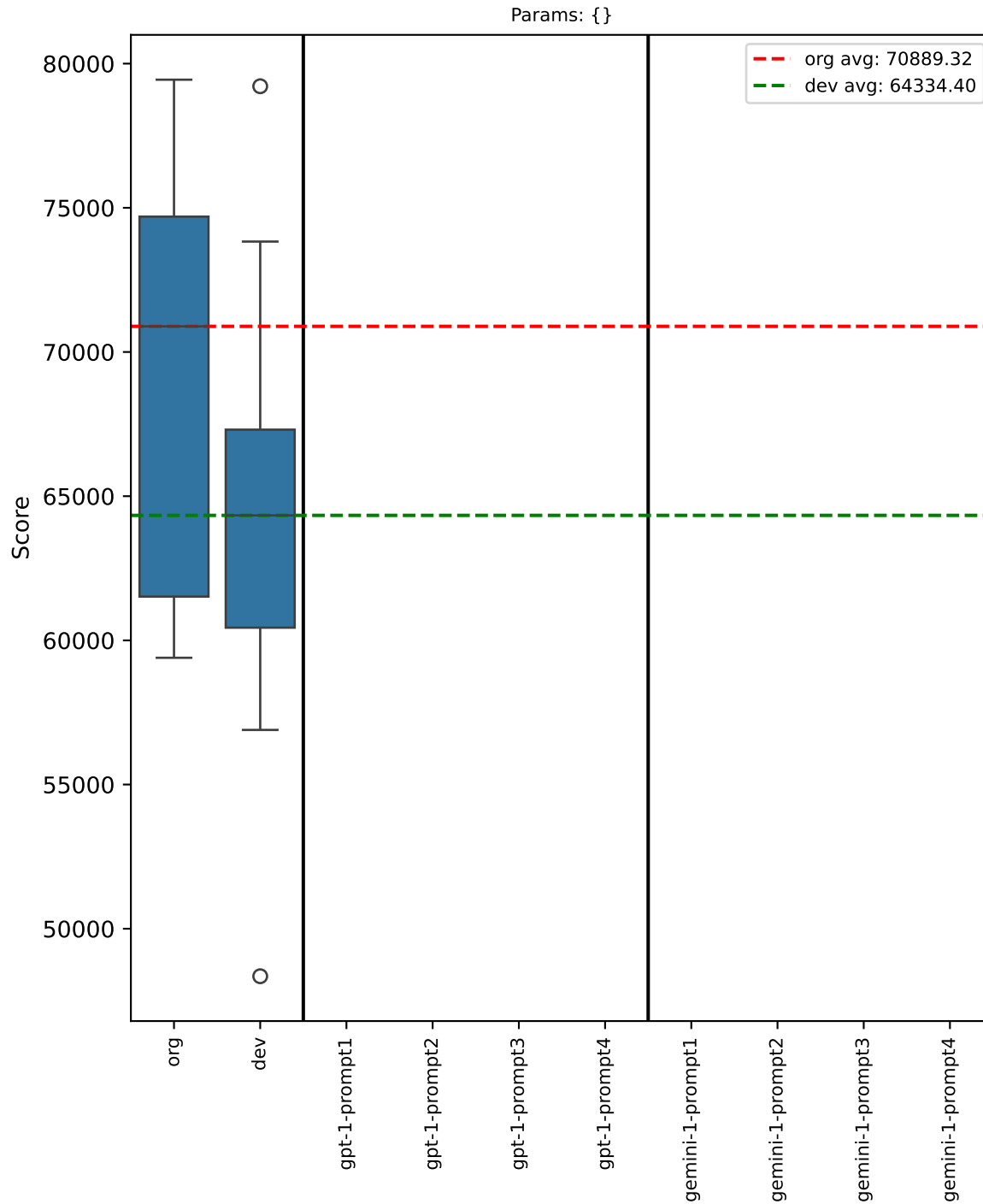
Benchmark: io.netty.microbench.buffer.UnsafeByteBufferBenchmark.setLongUnsafeByteBuffer



Benchmark: io.netty.microbench.channel.epoll.EpollSocketChannelBenchmark.executeMulti

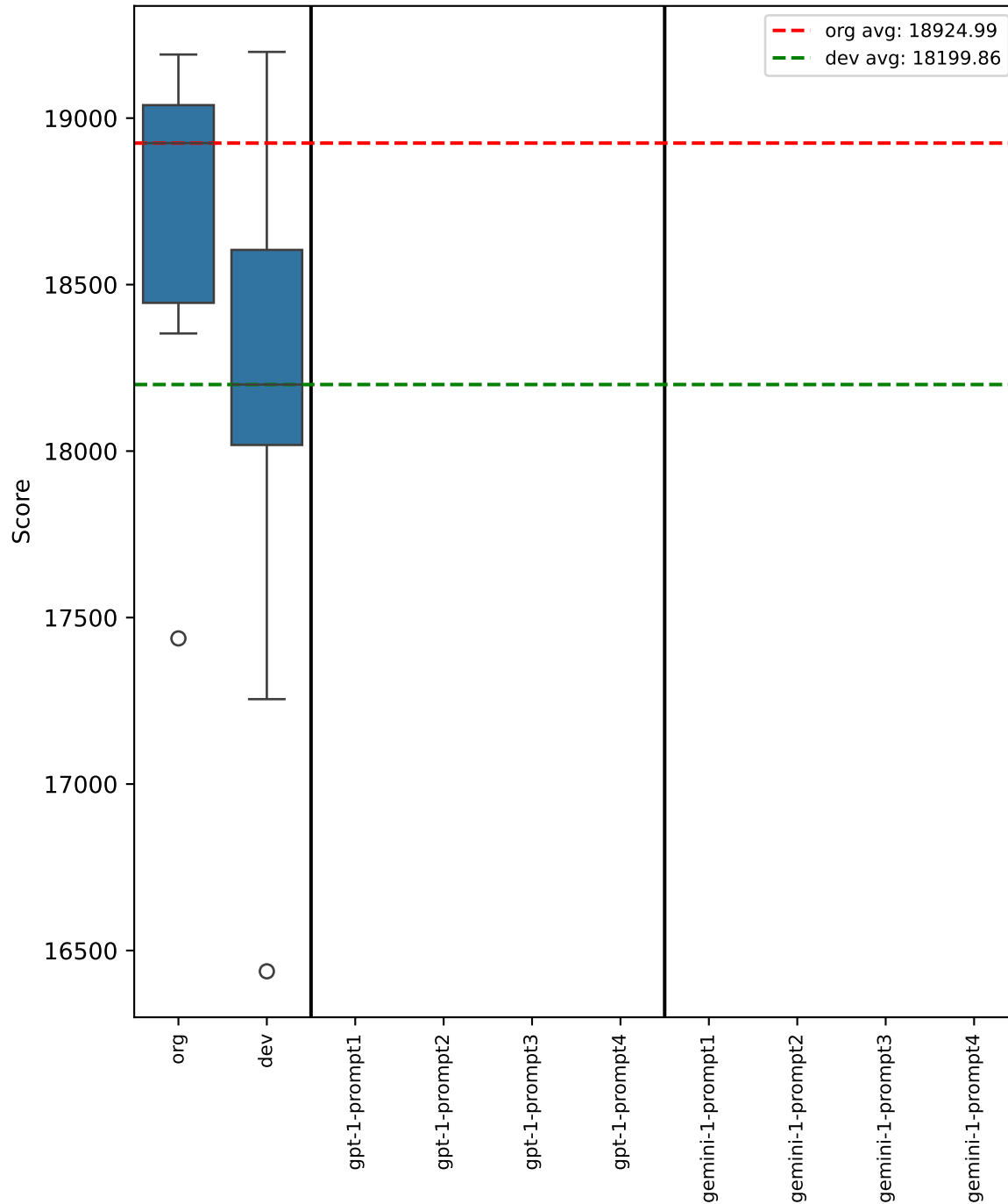


Benchmark: io.netty.microbench.channel.epoll.EpollSocketChannelBenchmark.executeSingle

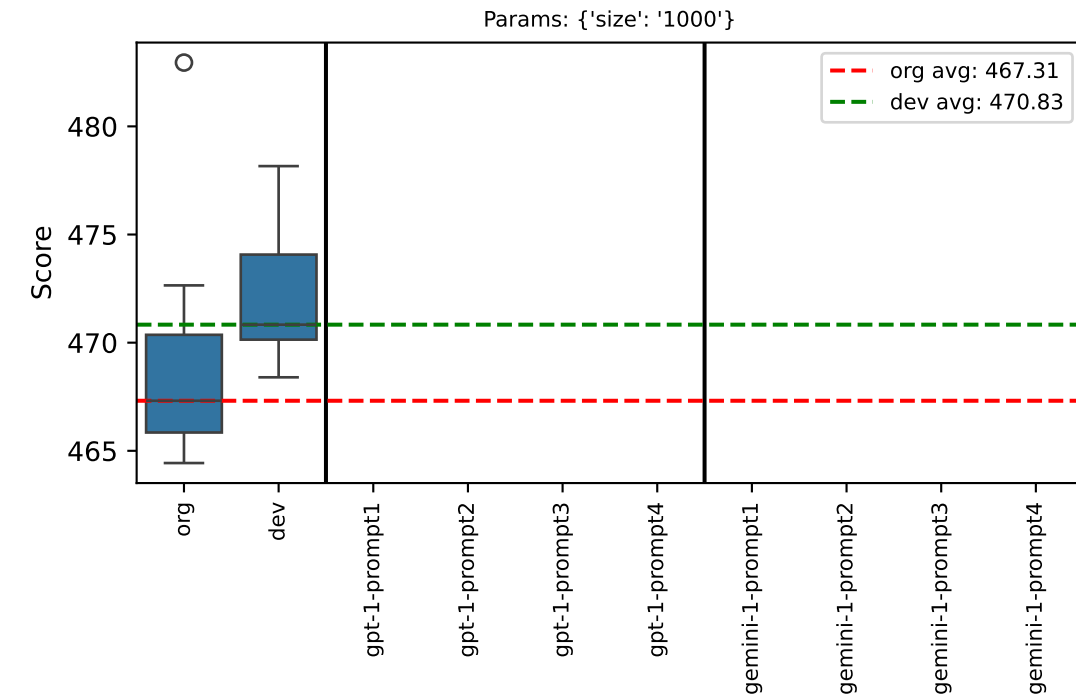
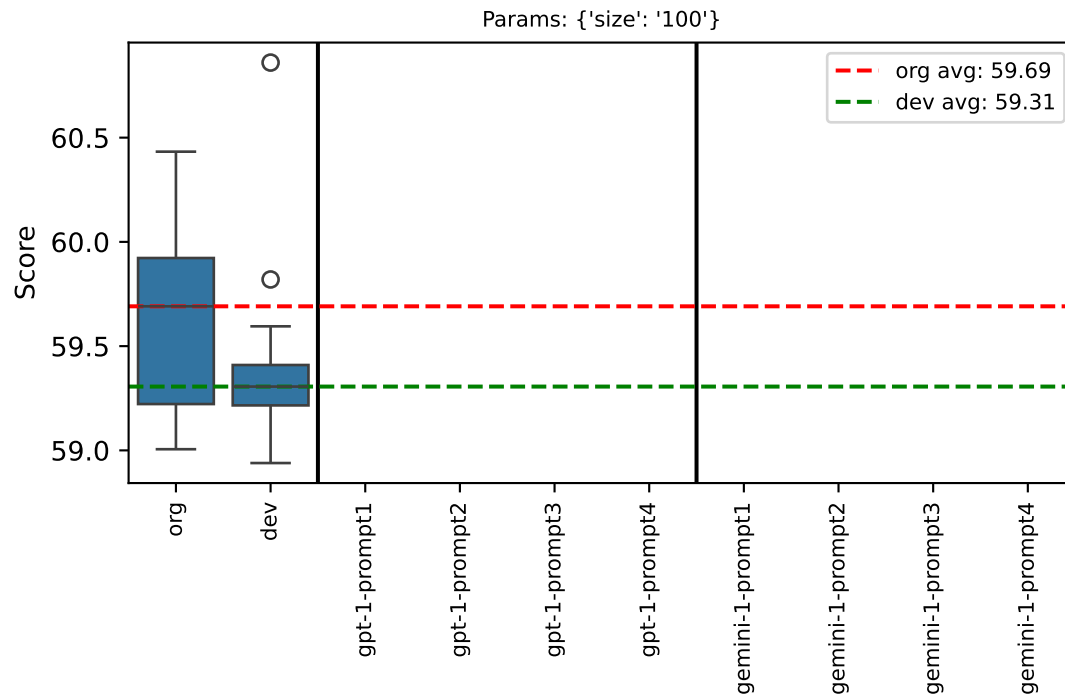
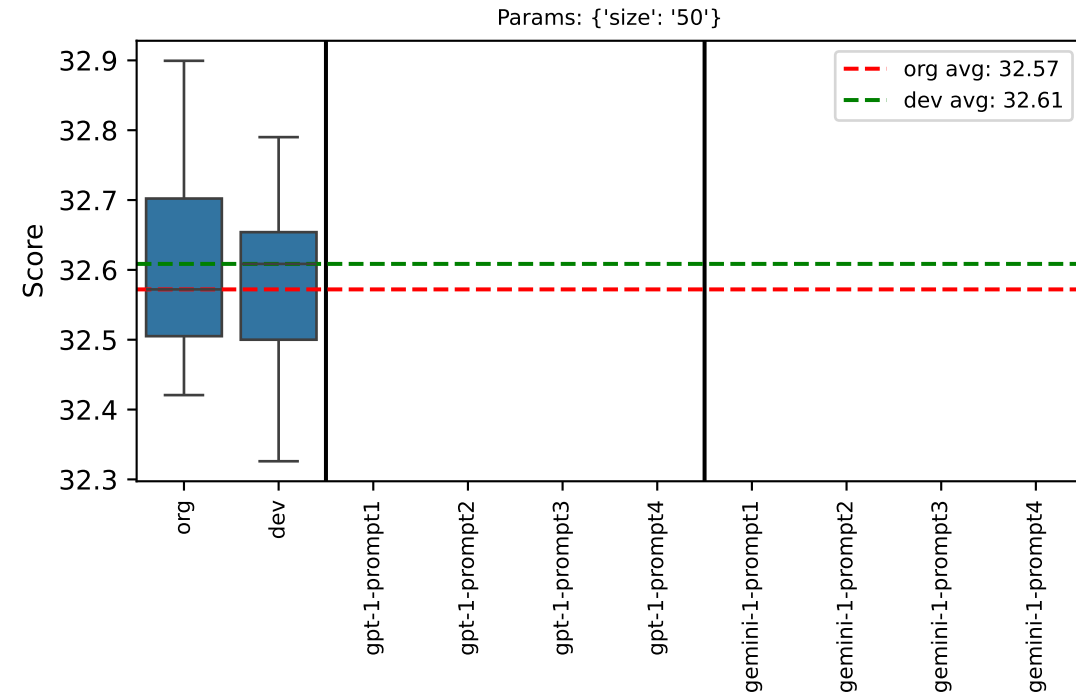
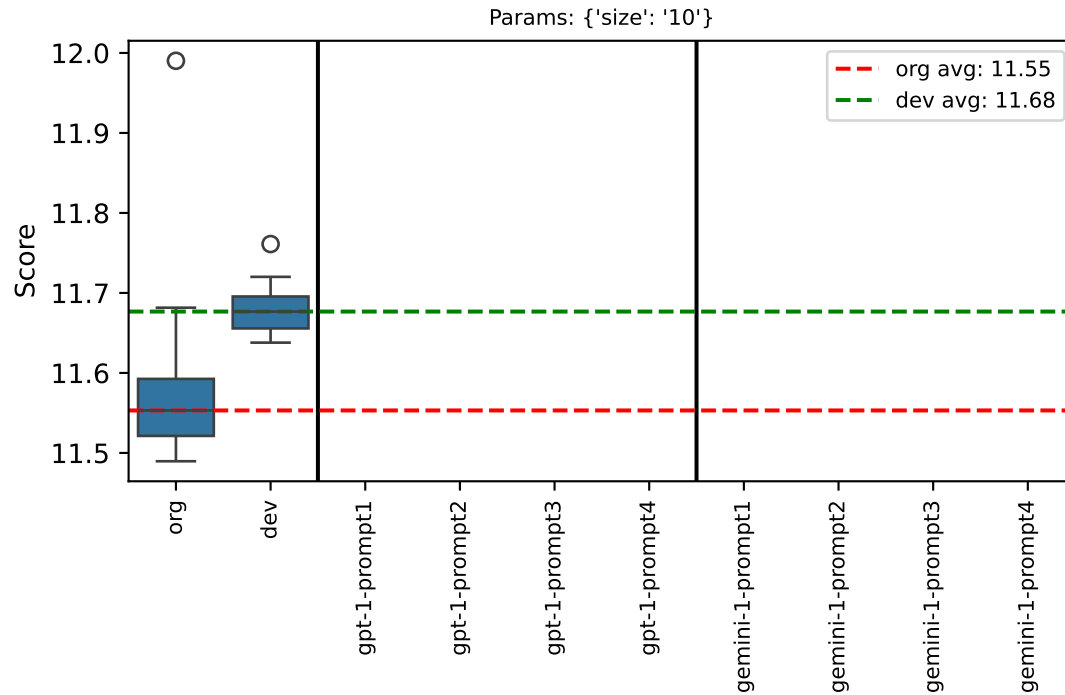


Benchmark: io.netty.microbench.channel.epoll.EpollSocketChannelBenchmark.pingPong

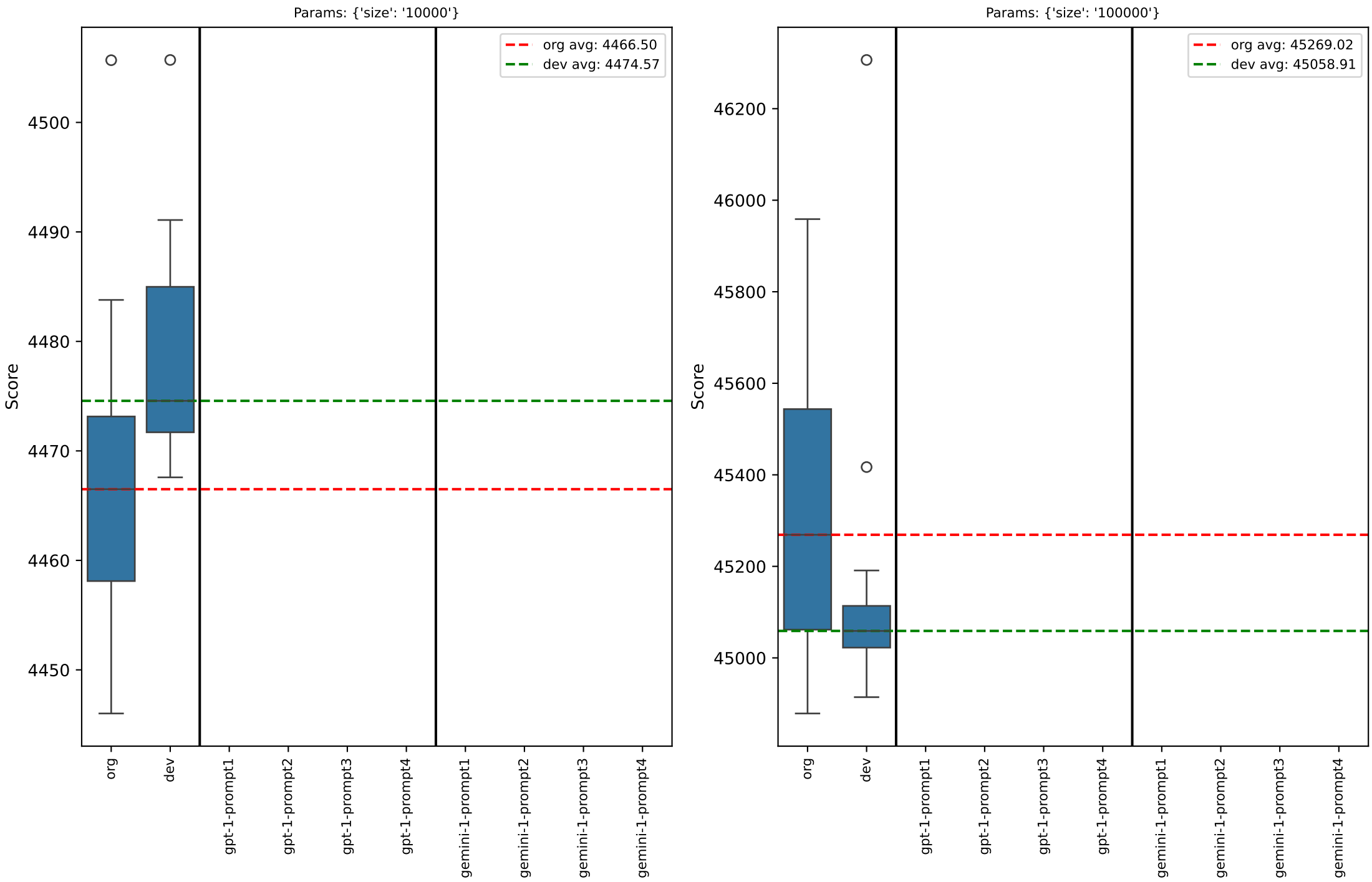
Params: {}



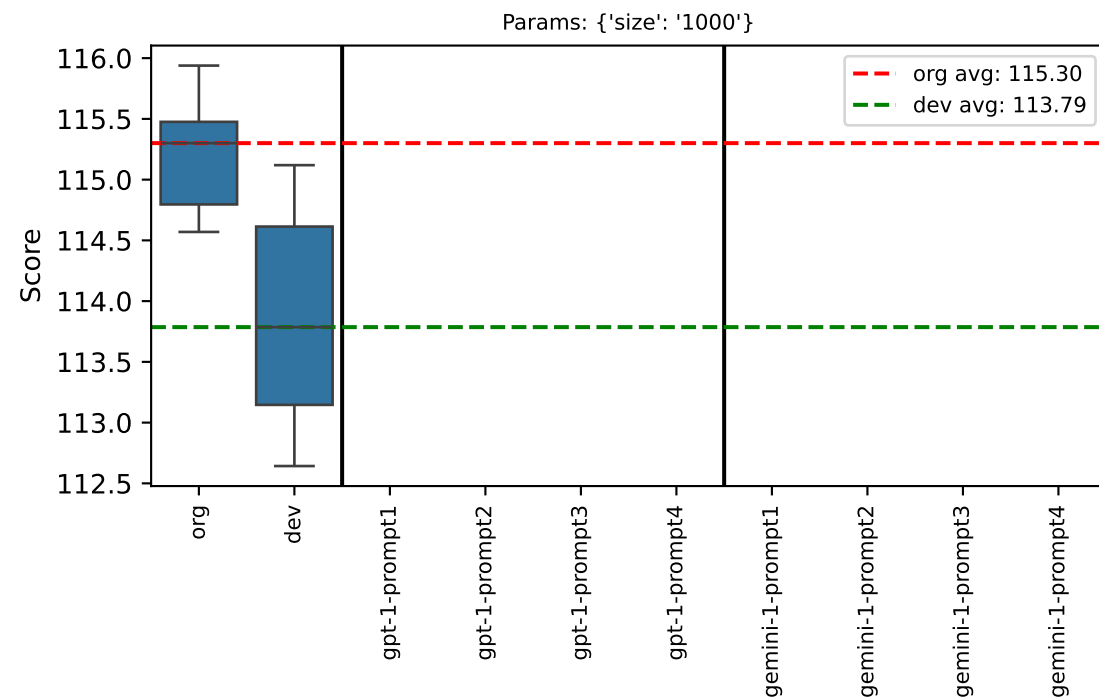
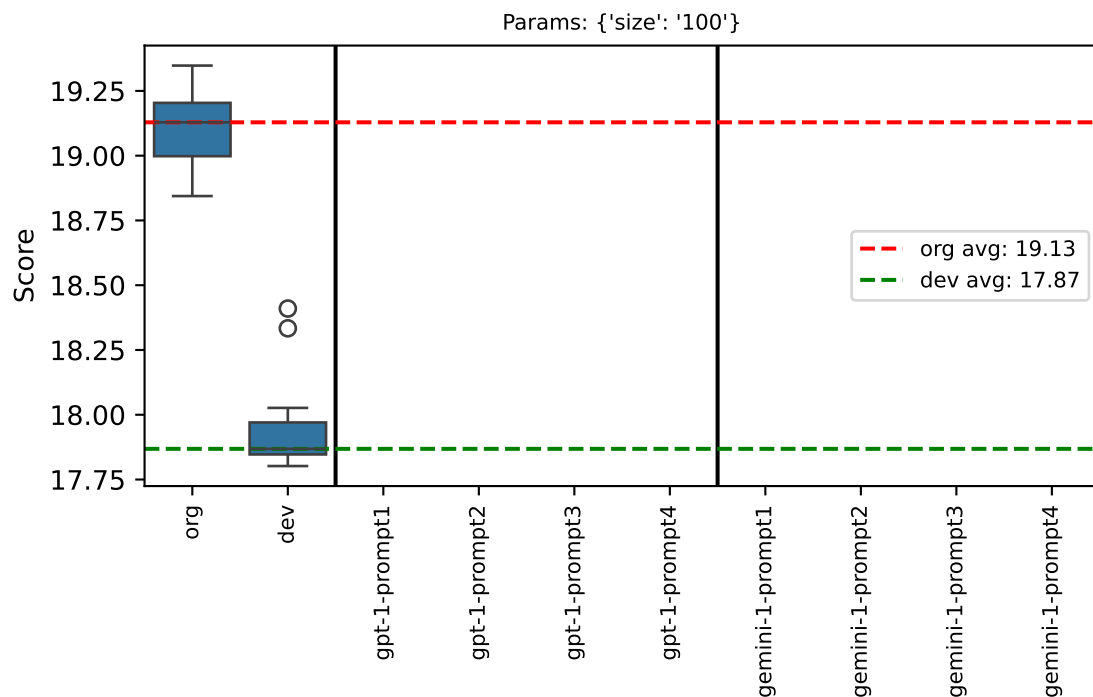
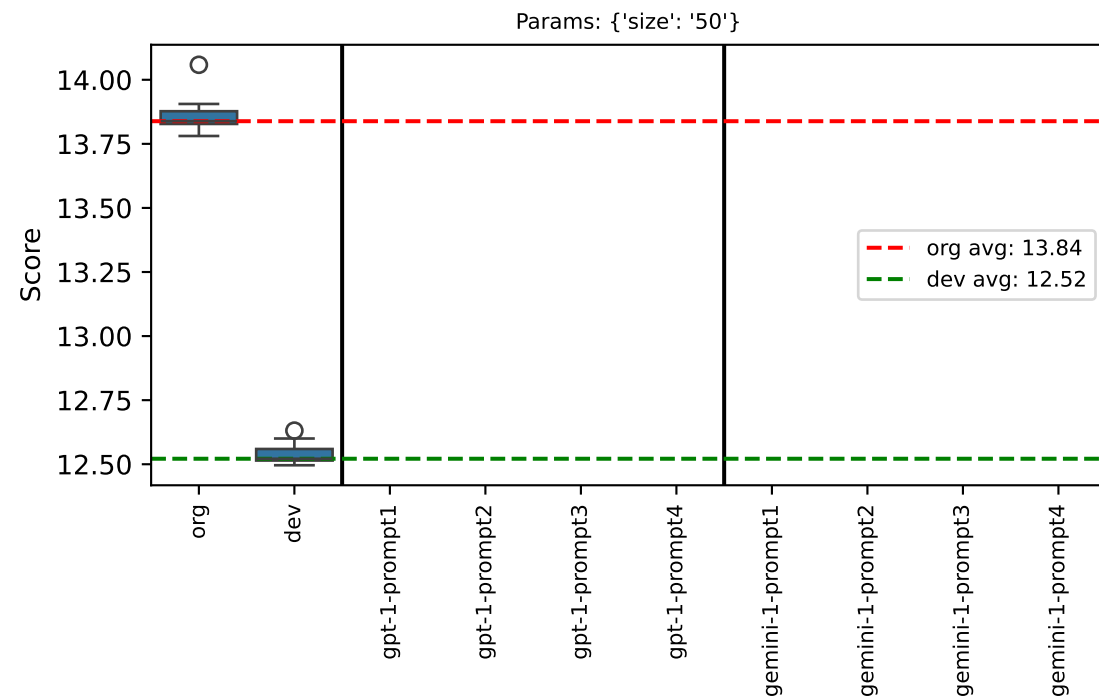
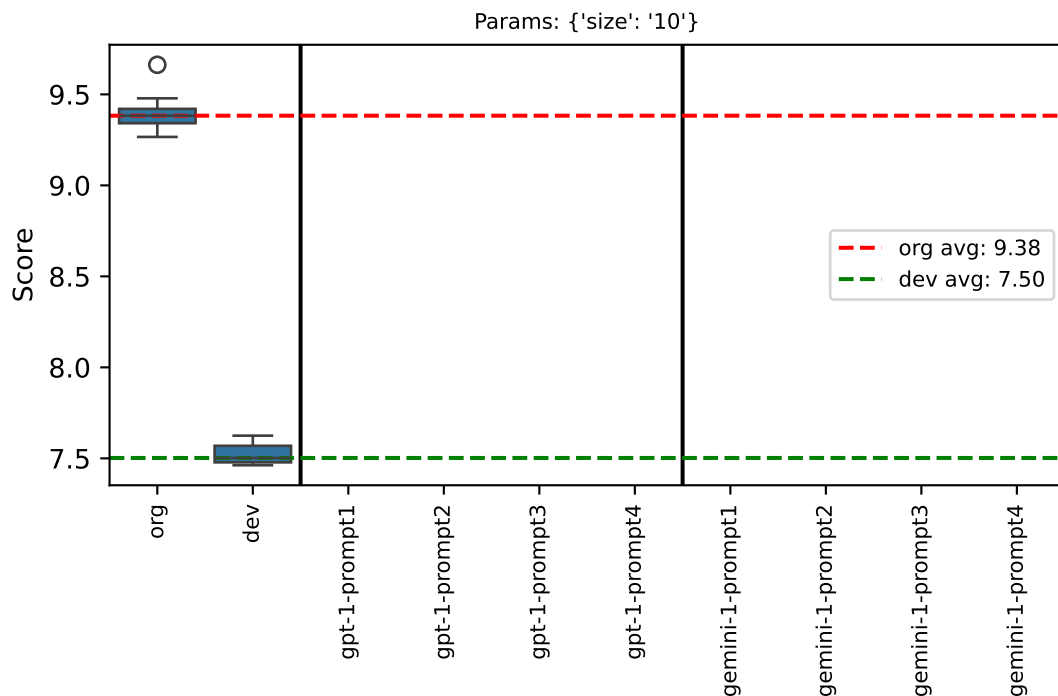
Benchmark: io.netty.microbench.internal.PlatformDependentBenchmark.arrayBytesEqual



Benchmark: io.netty.microbench.internal.PlatformDependentBenchmark.arrayBytesEqual

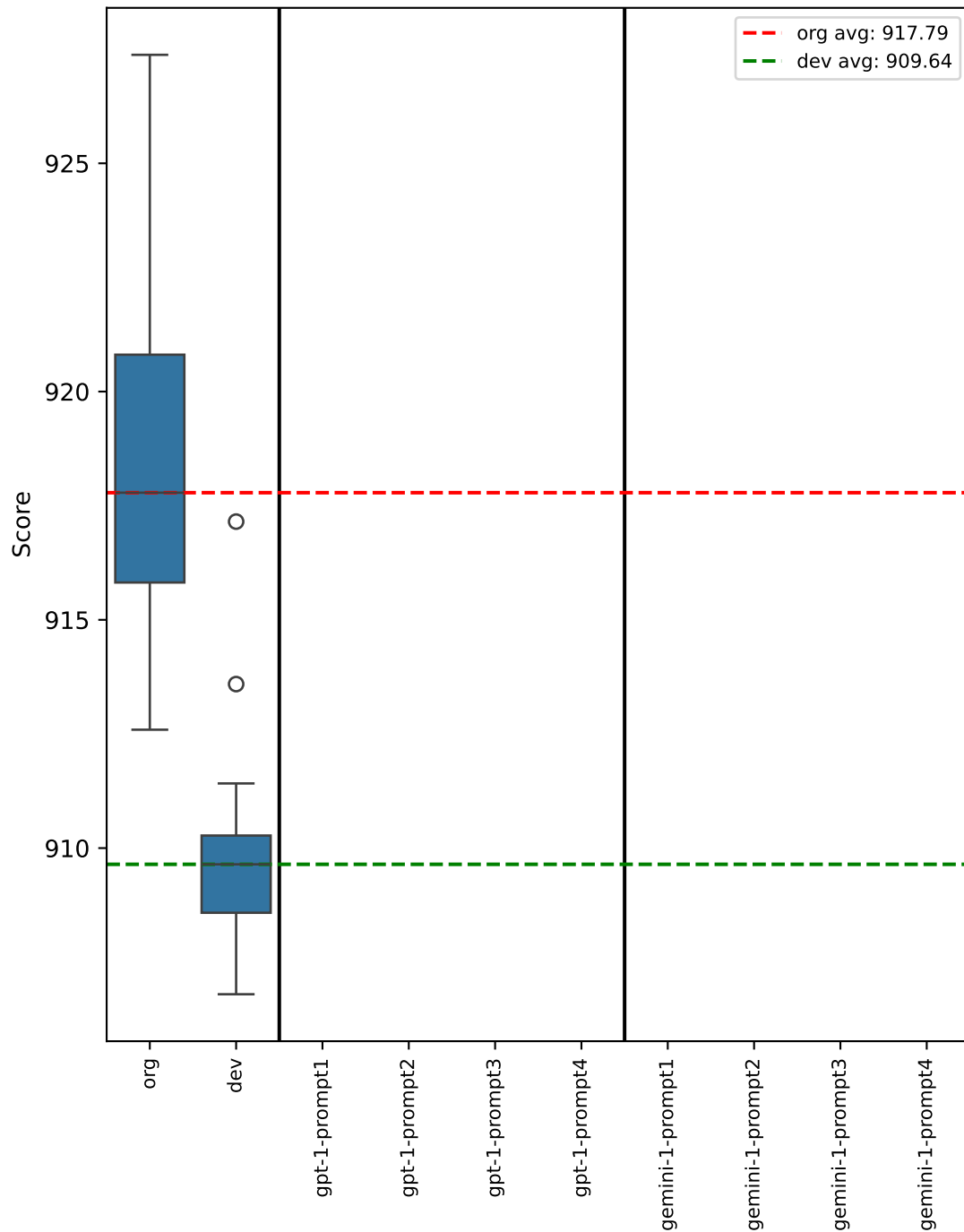


Benchmark: io.netty.microbench.internal.PlatformDependentBenchmark.unsafeBytesEqual

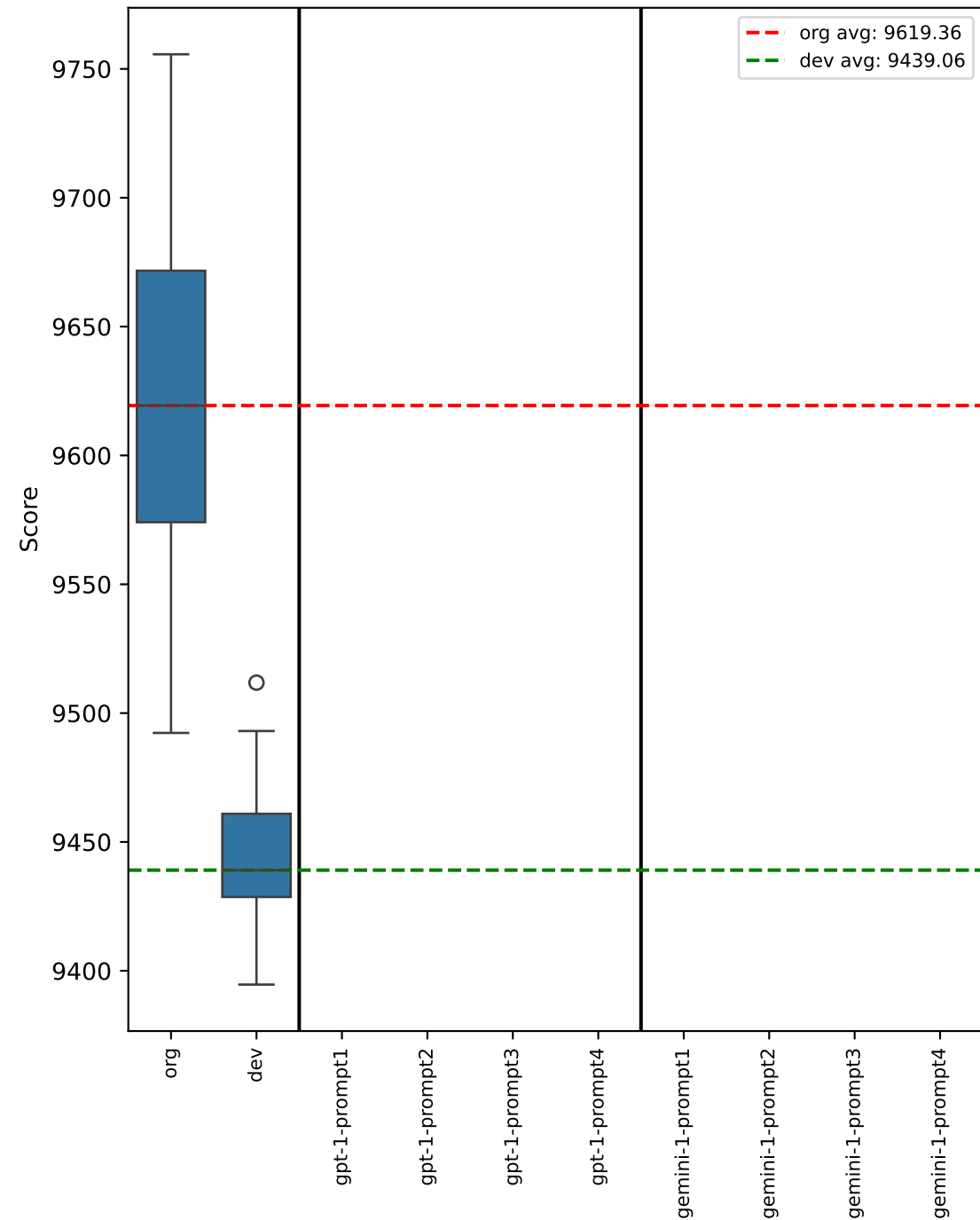


Benchmark: io.netty.microbench.internal.PlatformDependentBenchmark.unsafeBytesEqual

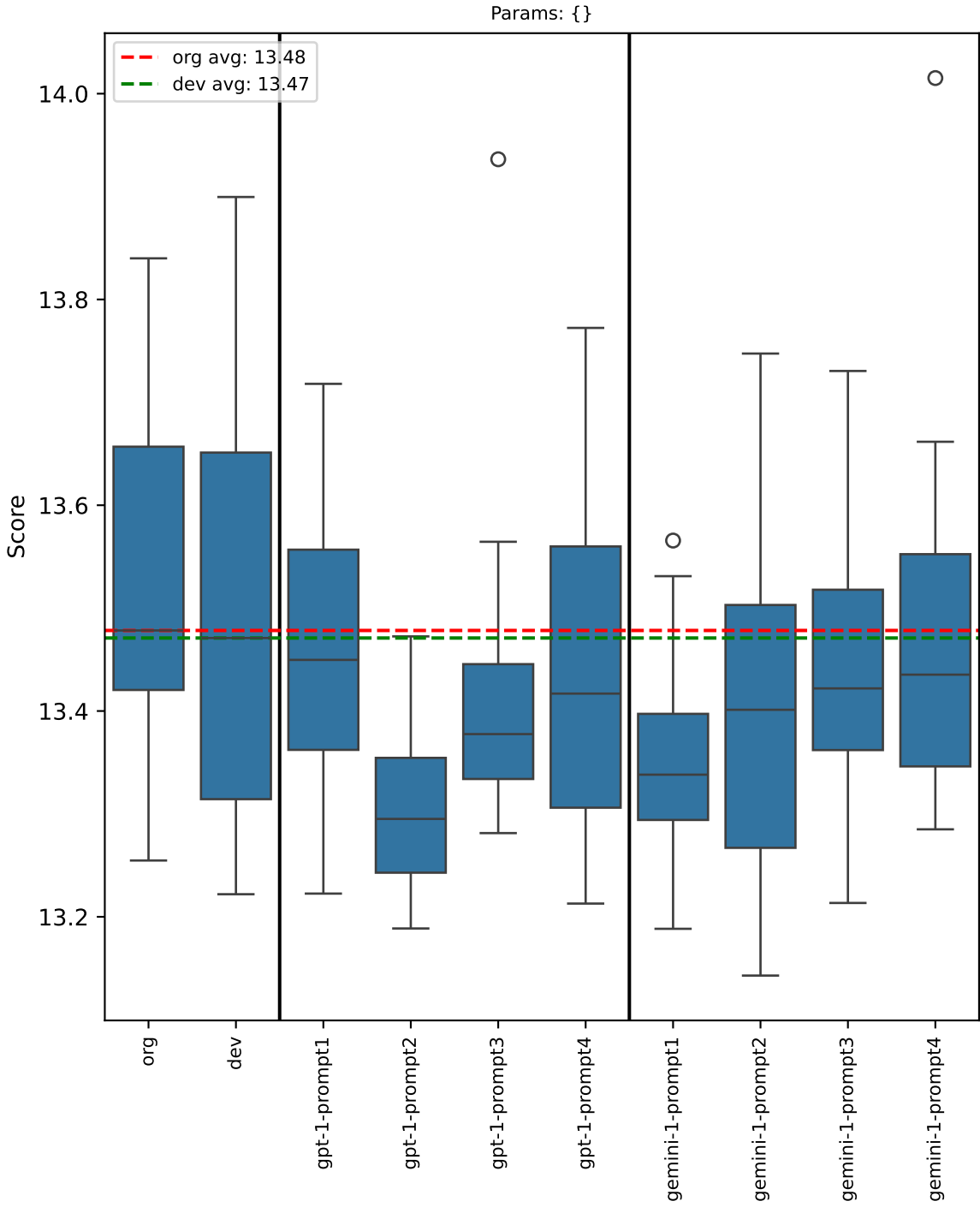
Params: {'size': '10000'}



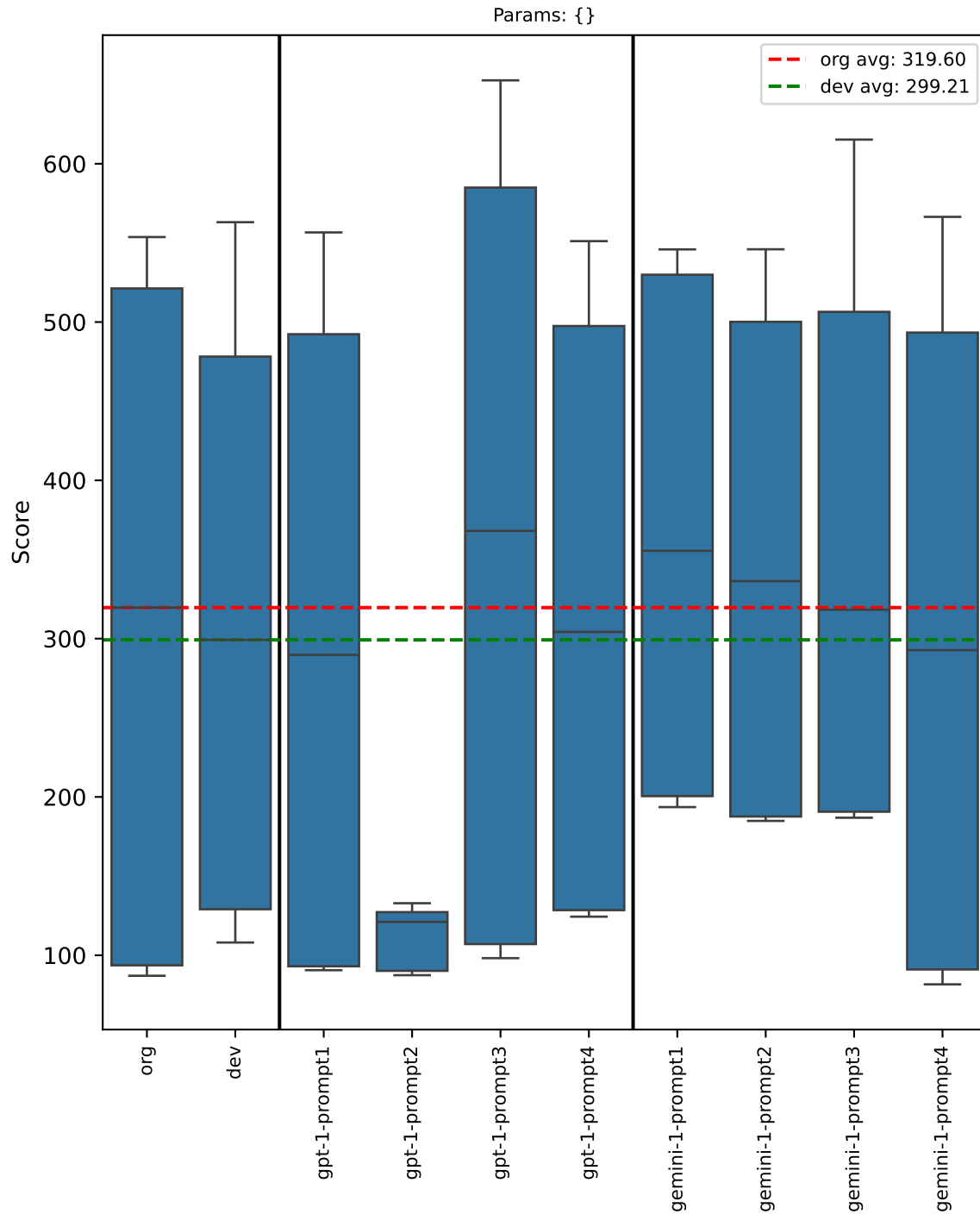
Params: {'size': '100000'}



Benchmark: io.netty.microbench.util.RecyclerBenchmark.plainNew

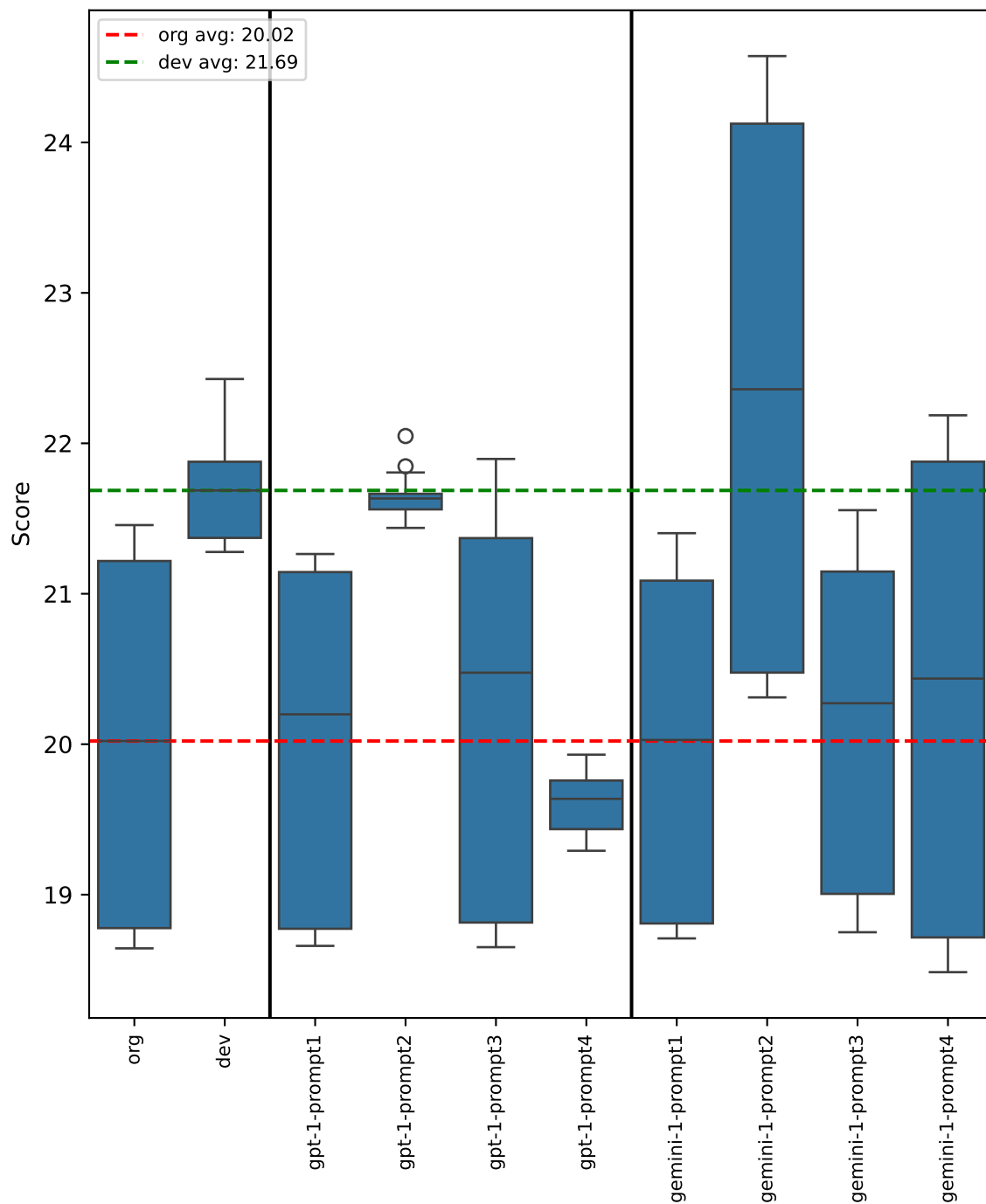


Benchmark: io.netty.microbench.util.RecyclerBenchmark.producerConsumer



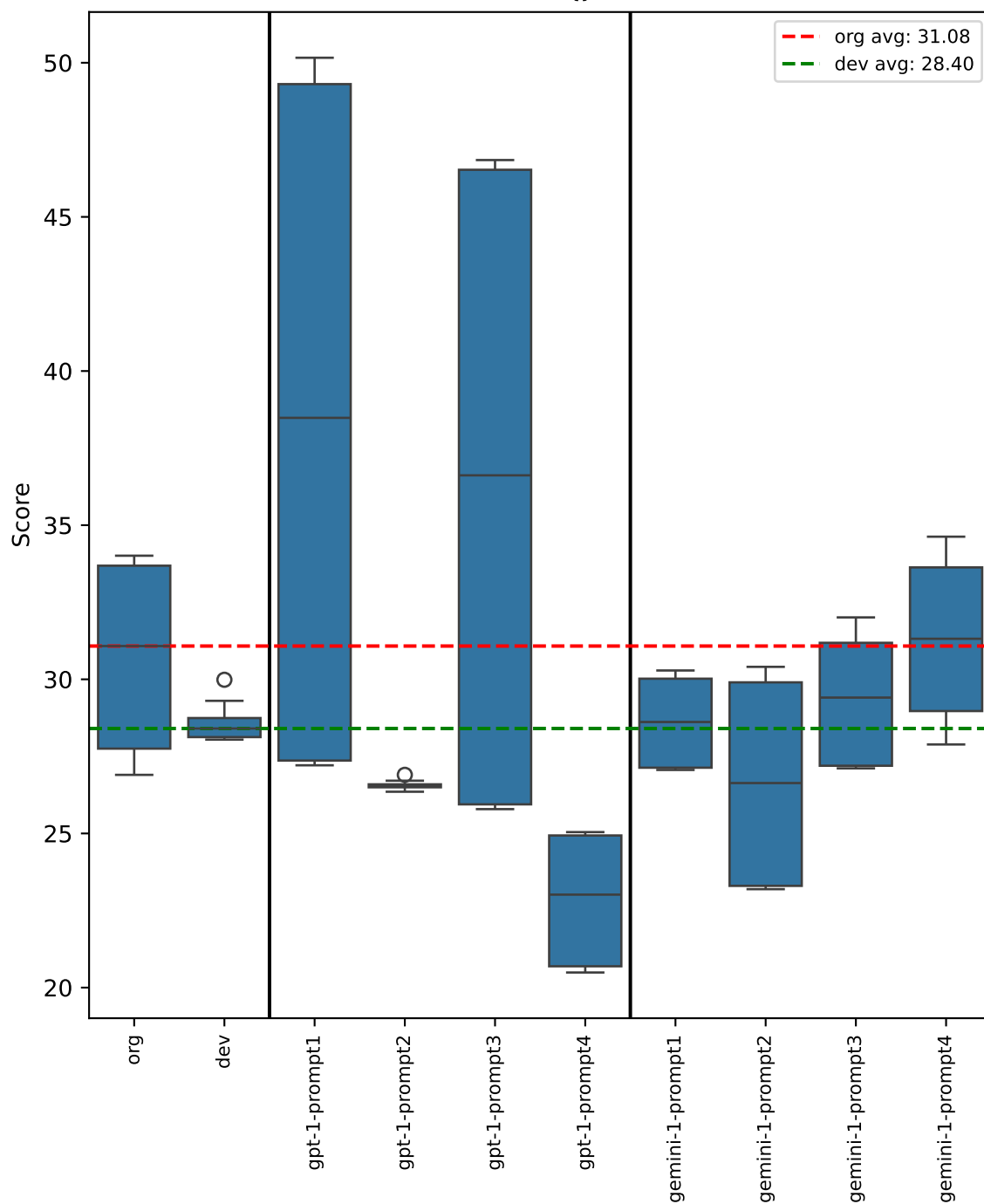
Benchmark: io.netty.microbench.util.RecyclerBenchmark.recyclerGetAndOrphan

Params: {}



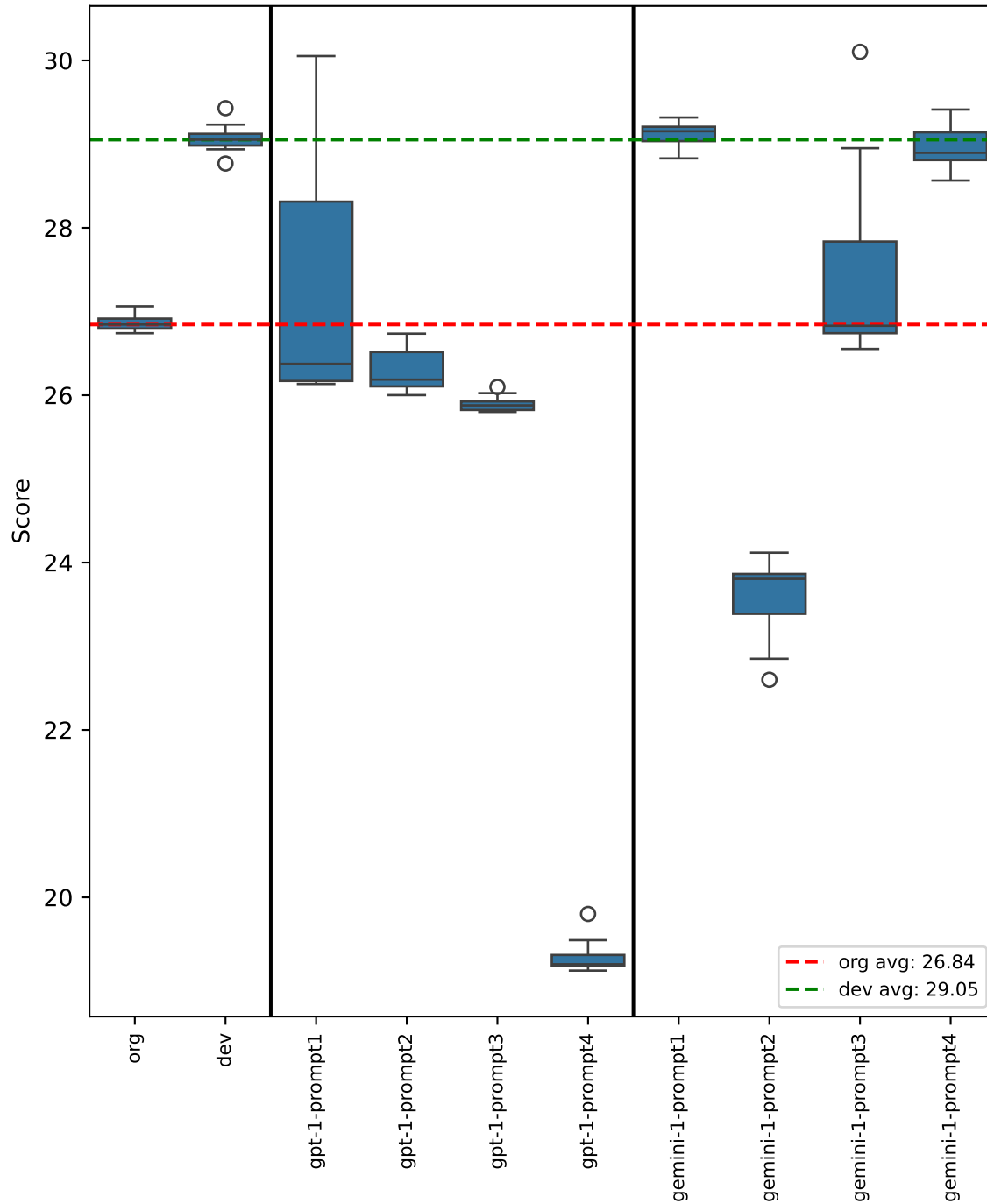
Benchmark: io.netty.microbench.util.RecyclerBenchmark.recyclerGetAndRecycle

Params: {}

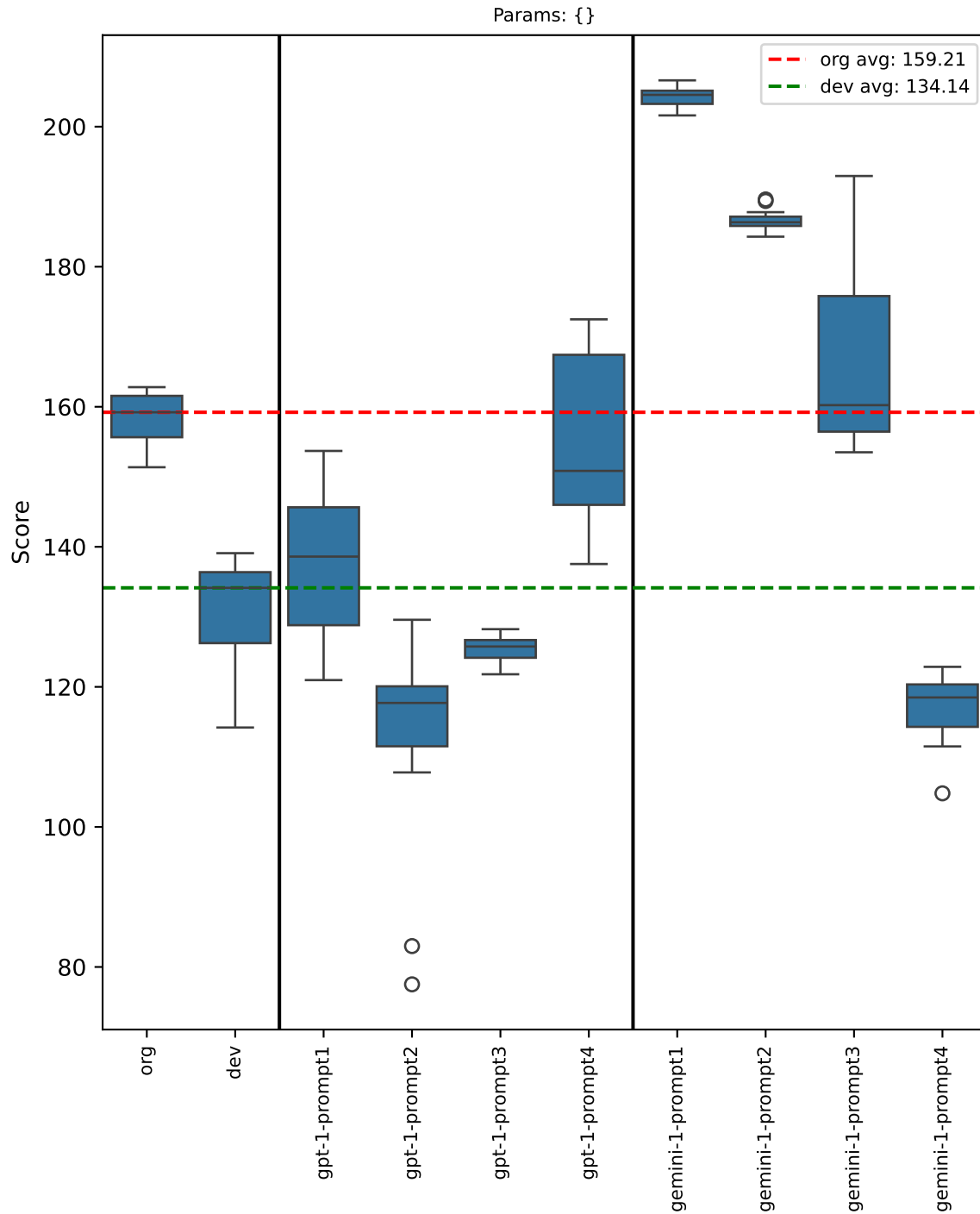


Benchmark: io.netty.microbench.util.RecyclerBenchmark.recyclerGetAndUnguardedRecycle

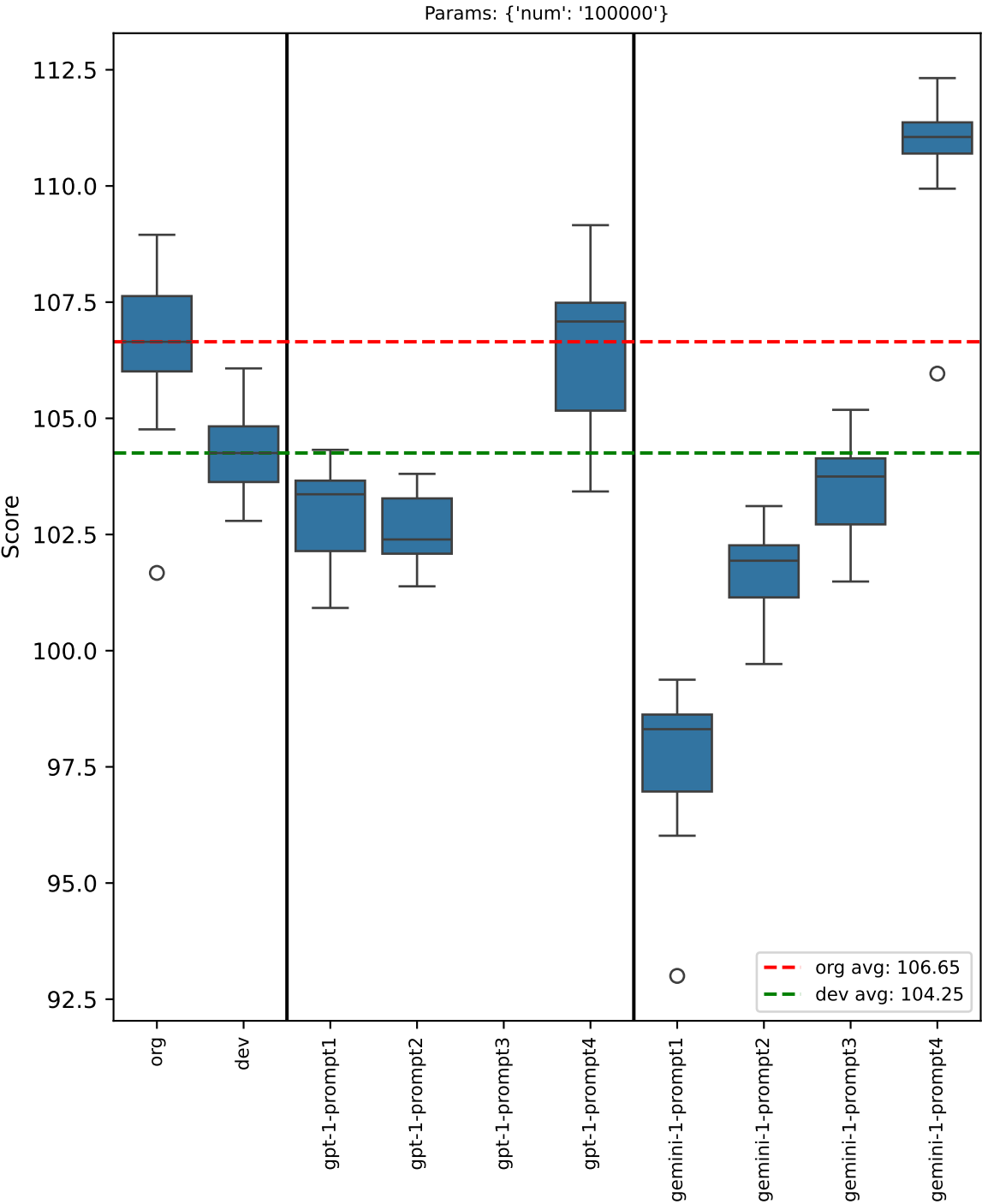
Params: {}



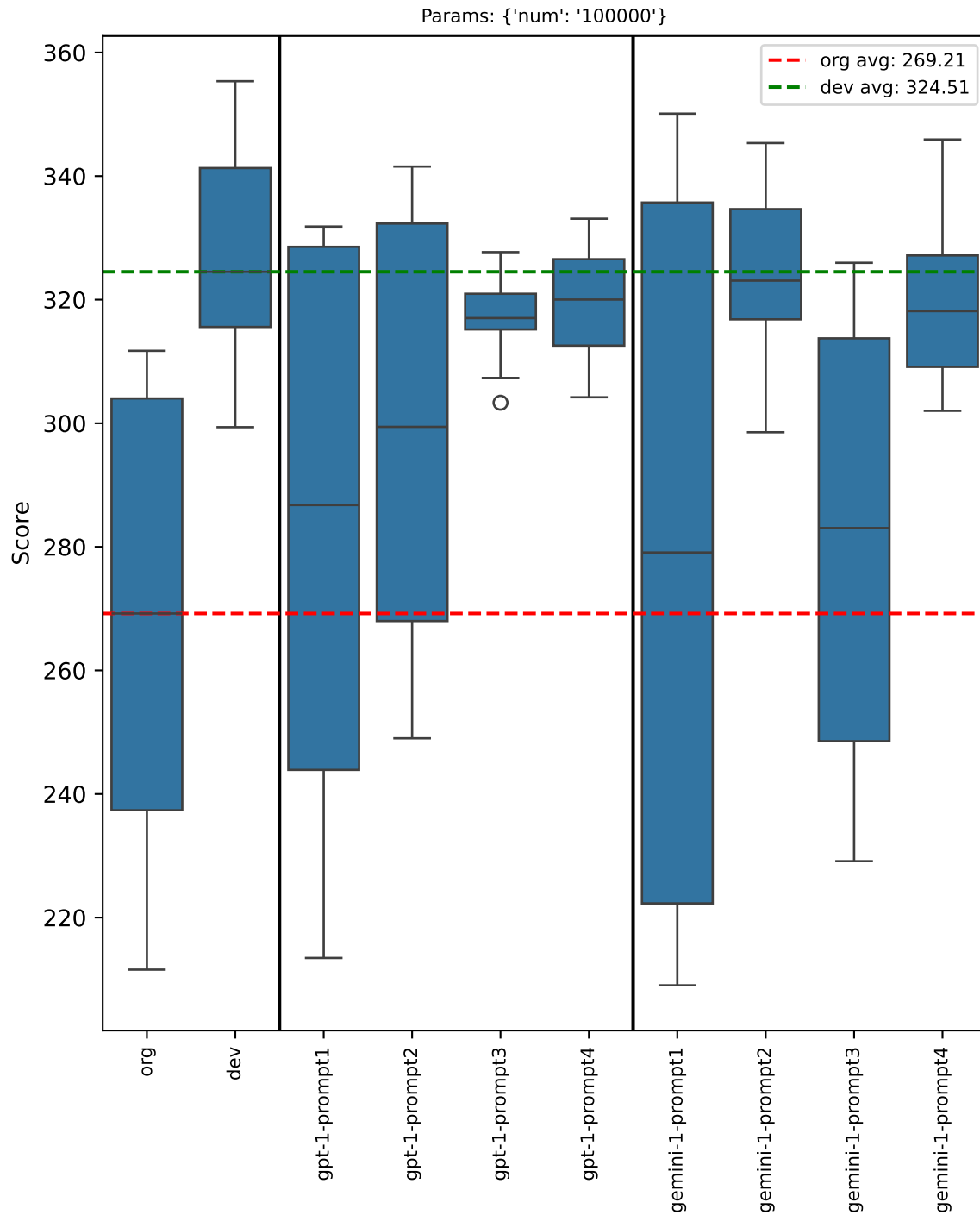
Benchmark: io.netty.microbench.util.RecyclerBenchmark.unguardedProducerConsumer



Benchmark: io.netty.util.concurrent.ScheduleFutureTaskBenchmark.scheduleCancelLotsOutsideLoop

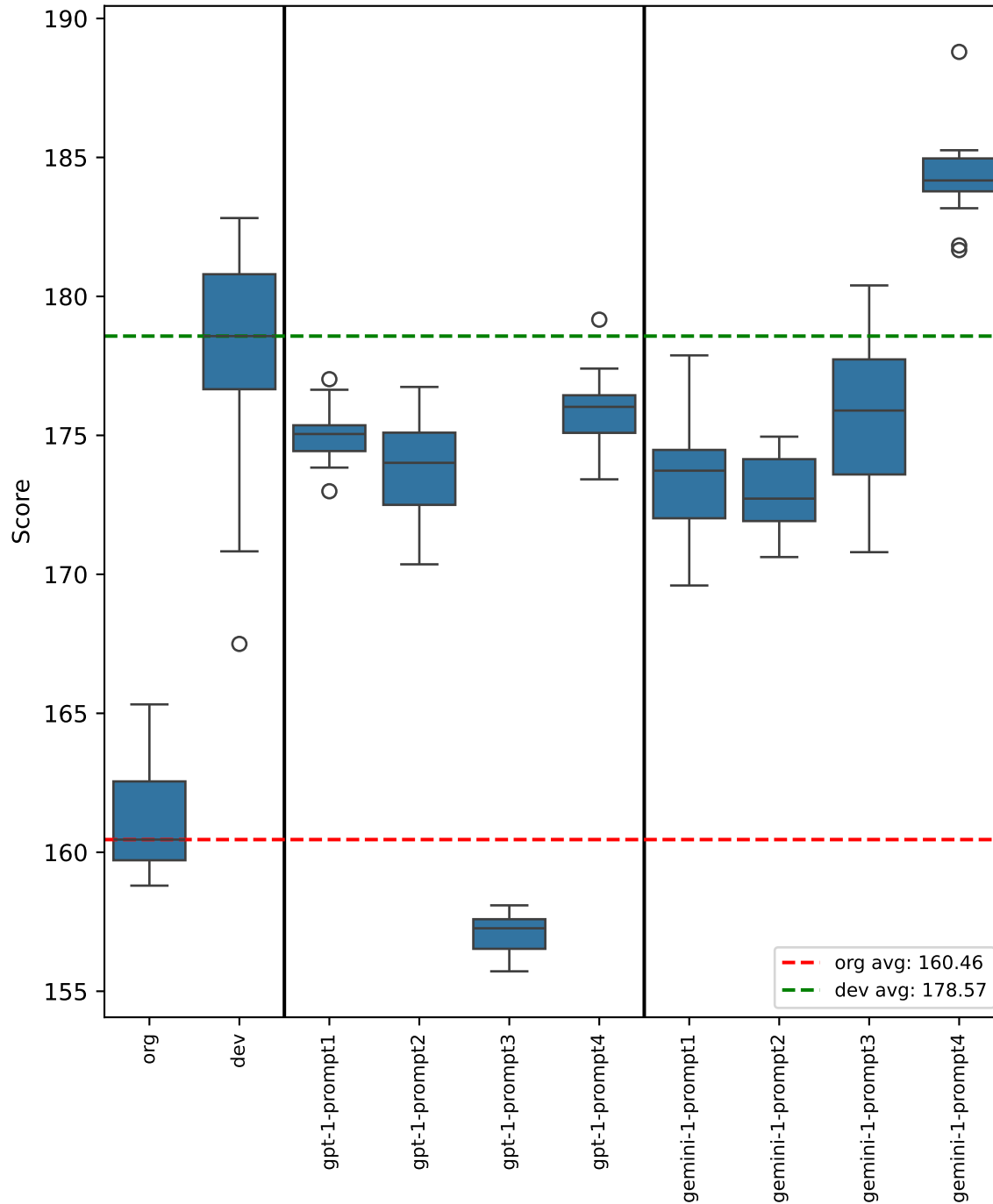


Benchmark: io.netty.util.concurrent.ScheduleFutureTaskBenchmark.scheduleLots

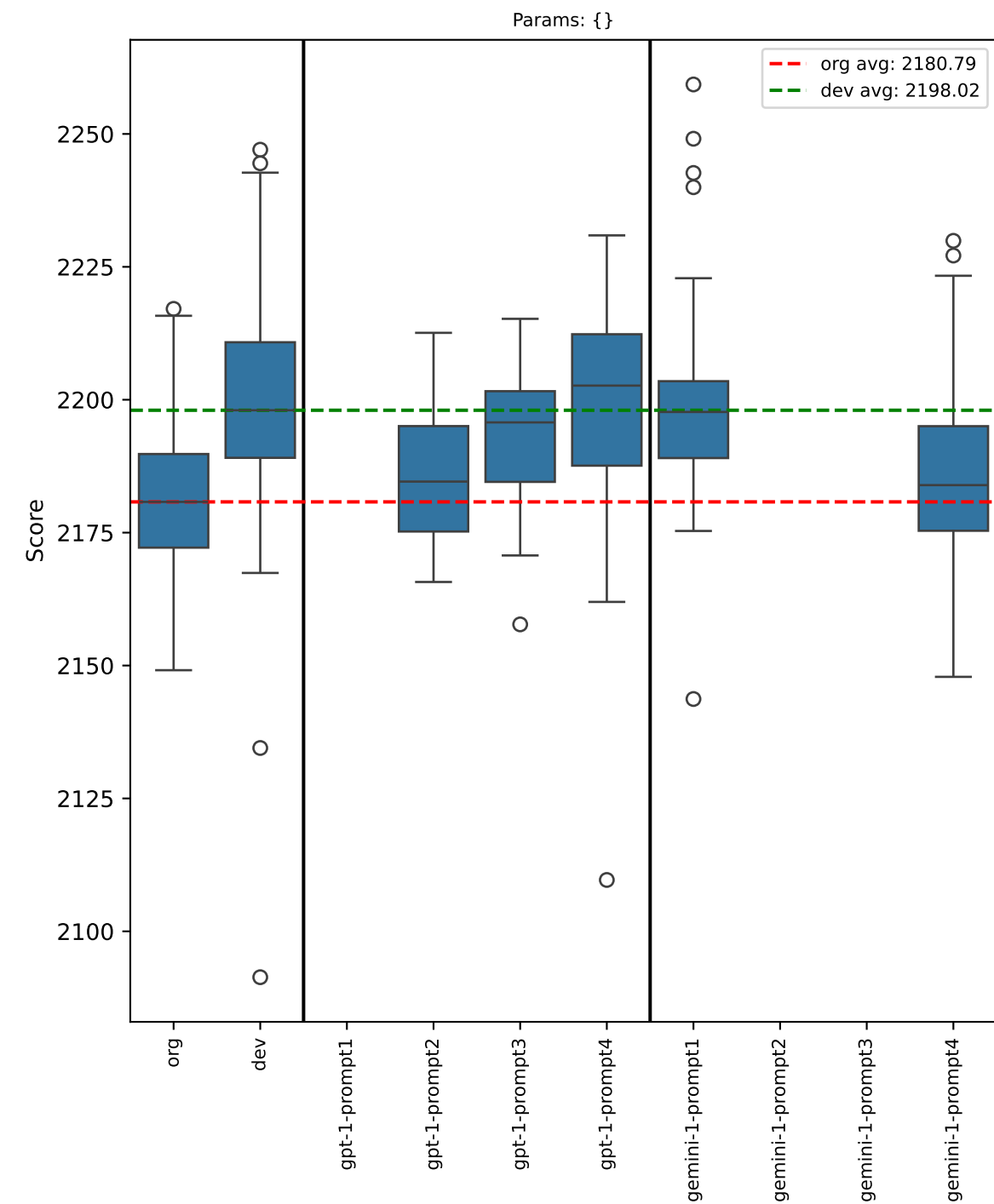


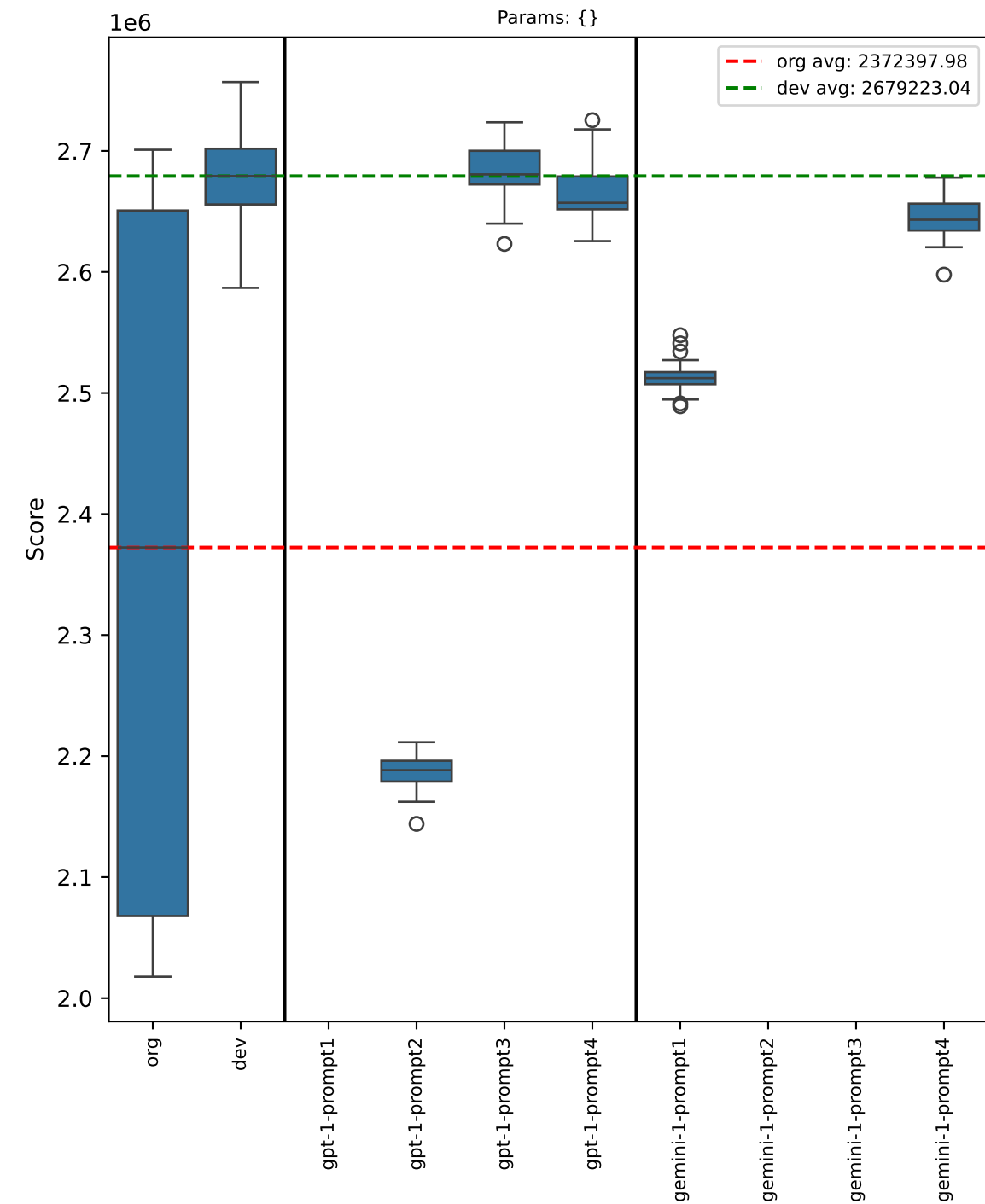
Benchmark: io.netty.util.concurrent.ScheduleFutureTaskBenchmark.scheduleLotsOutsideLoop

Params: {'num': '100000'}

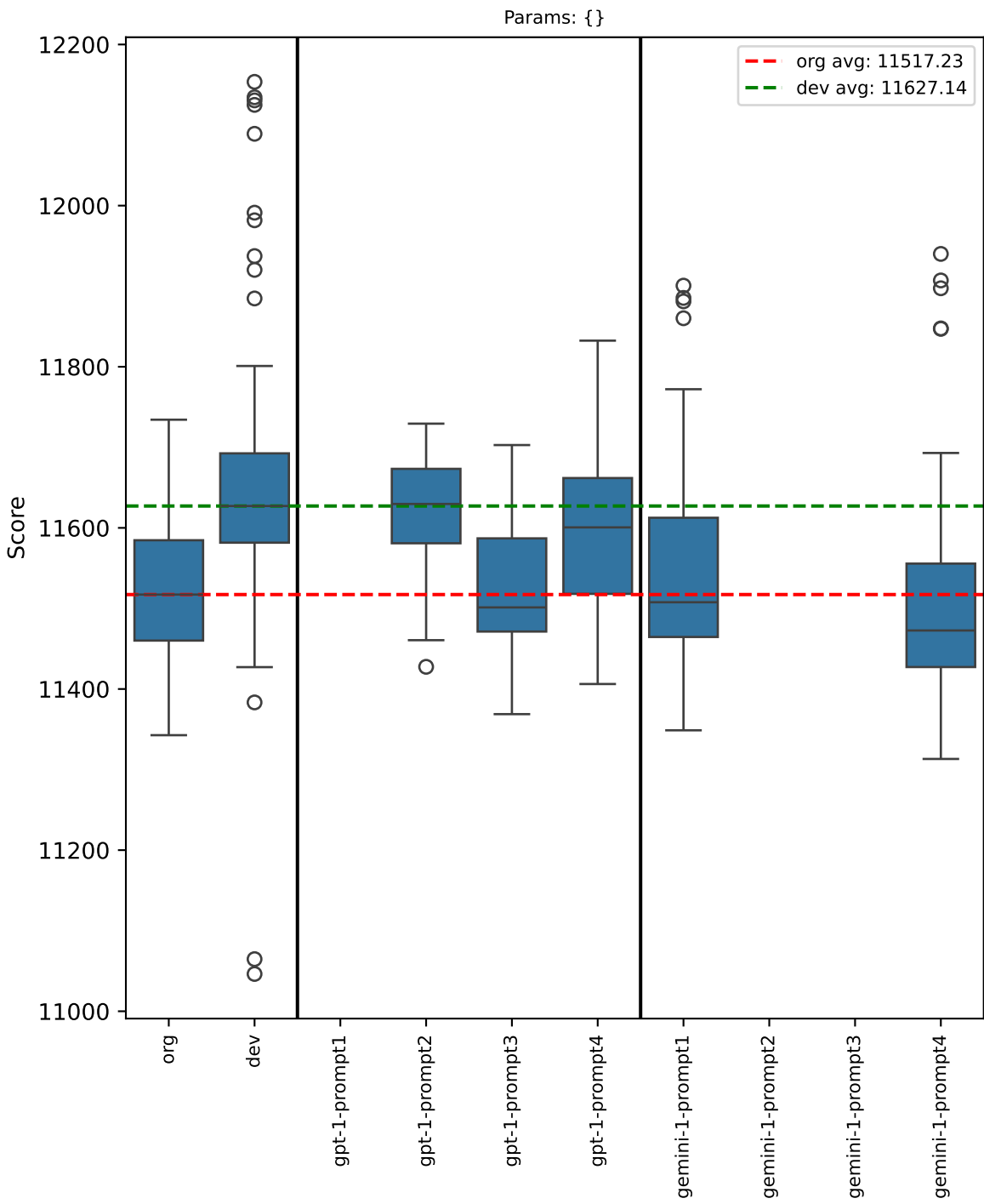


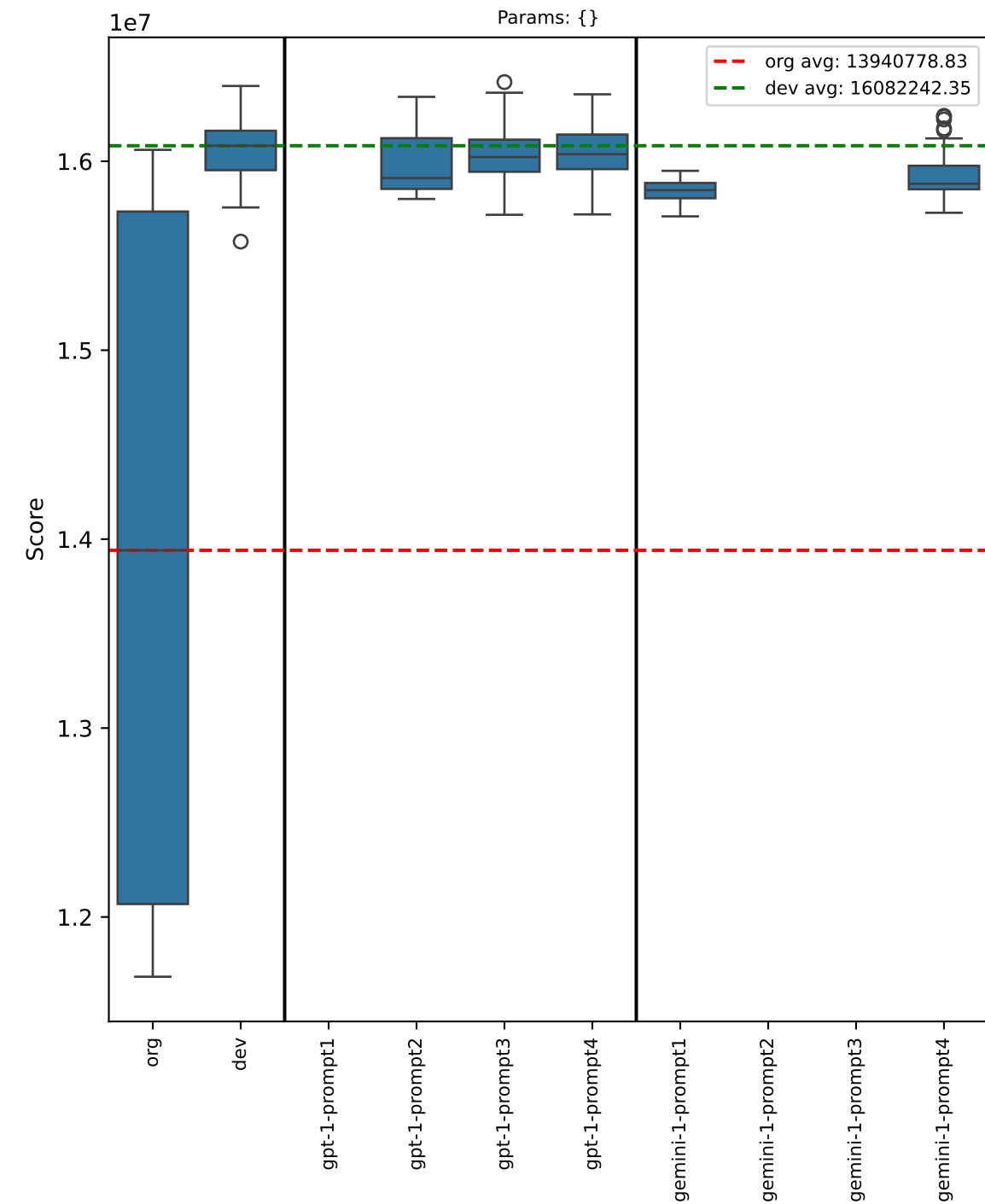
Project: presto

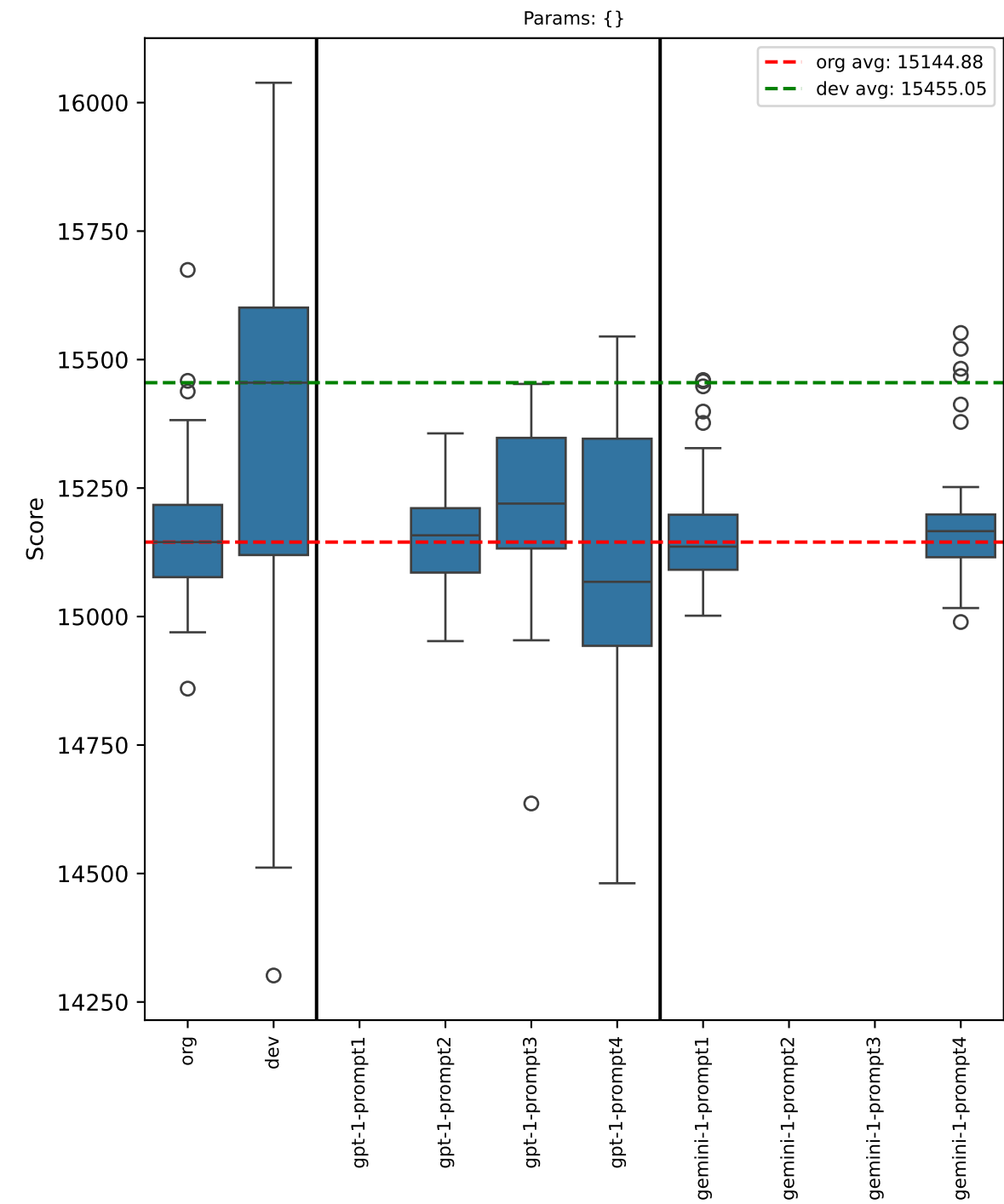


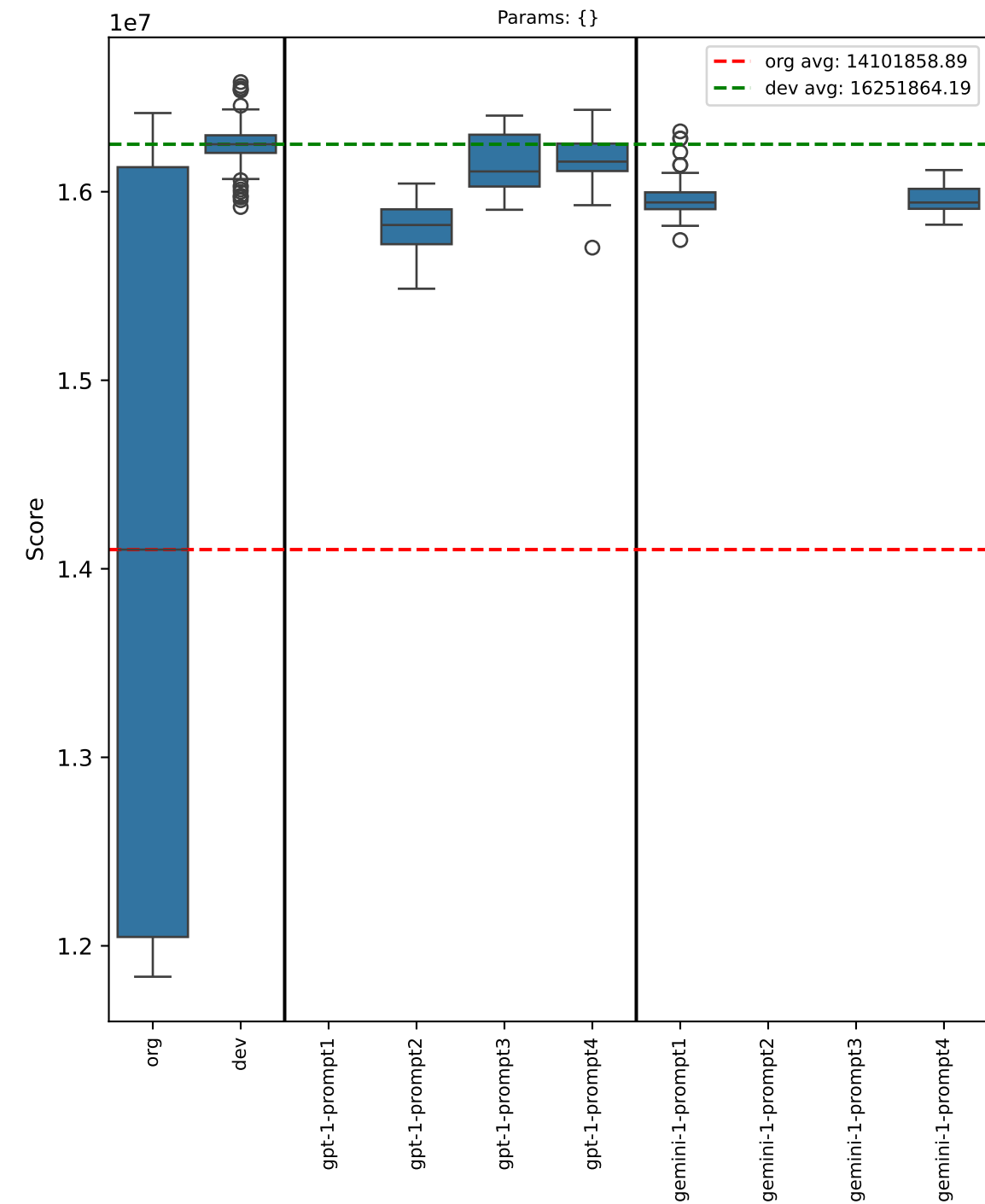


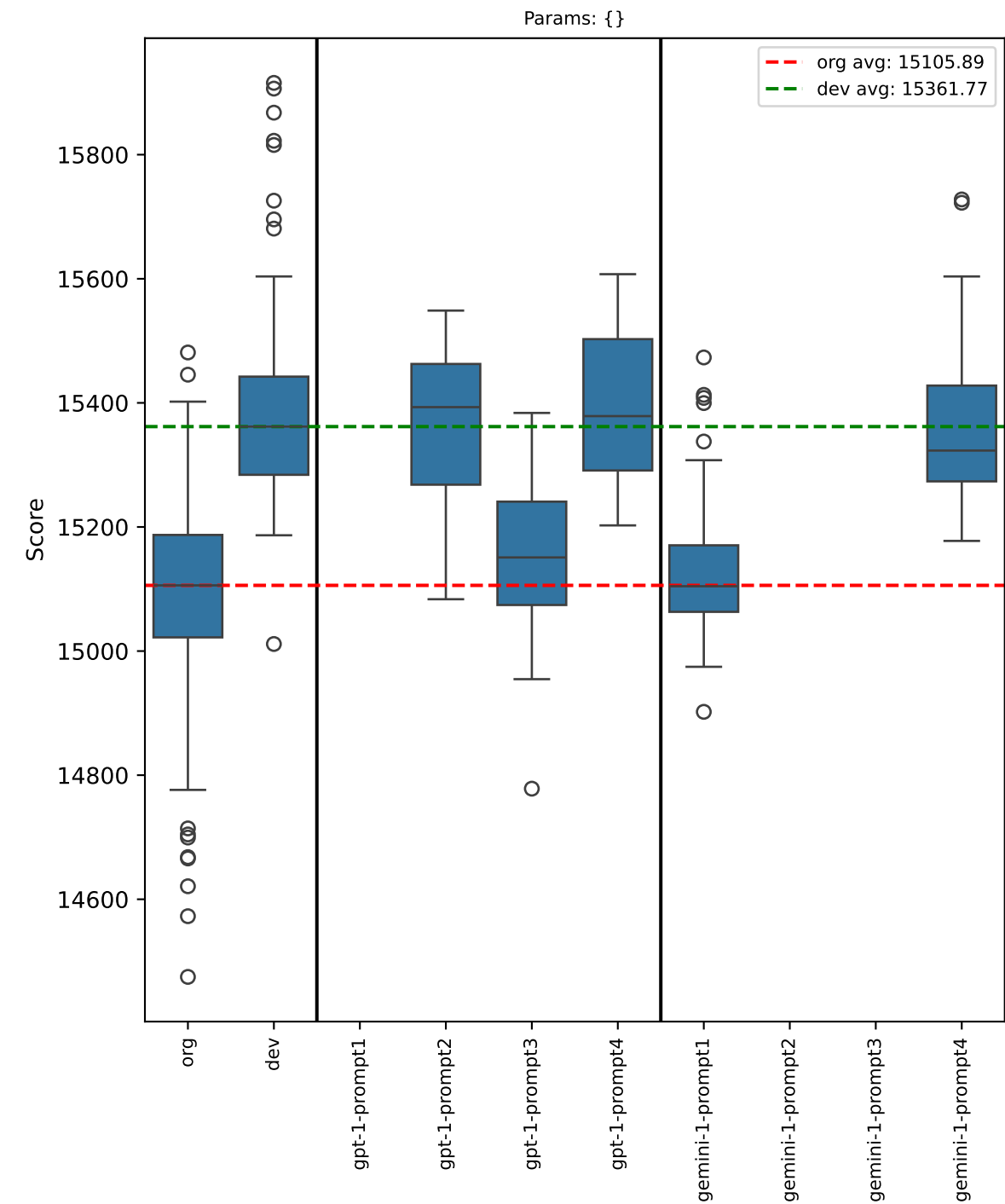
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeComplexLineString

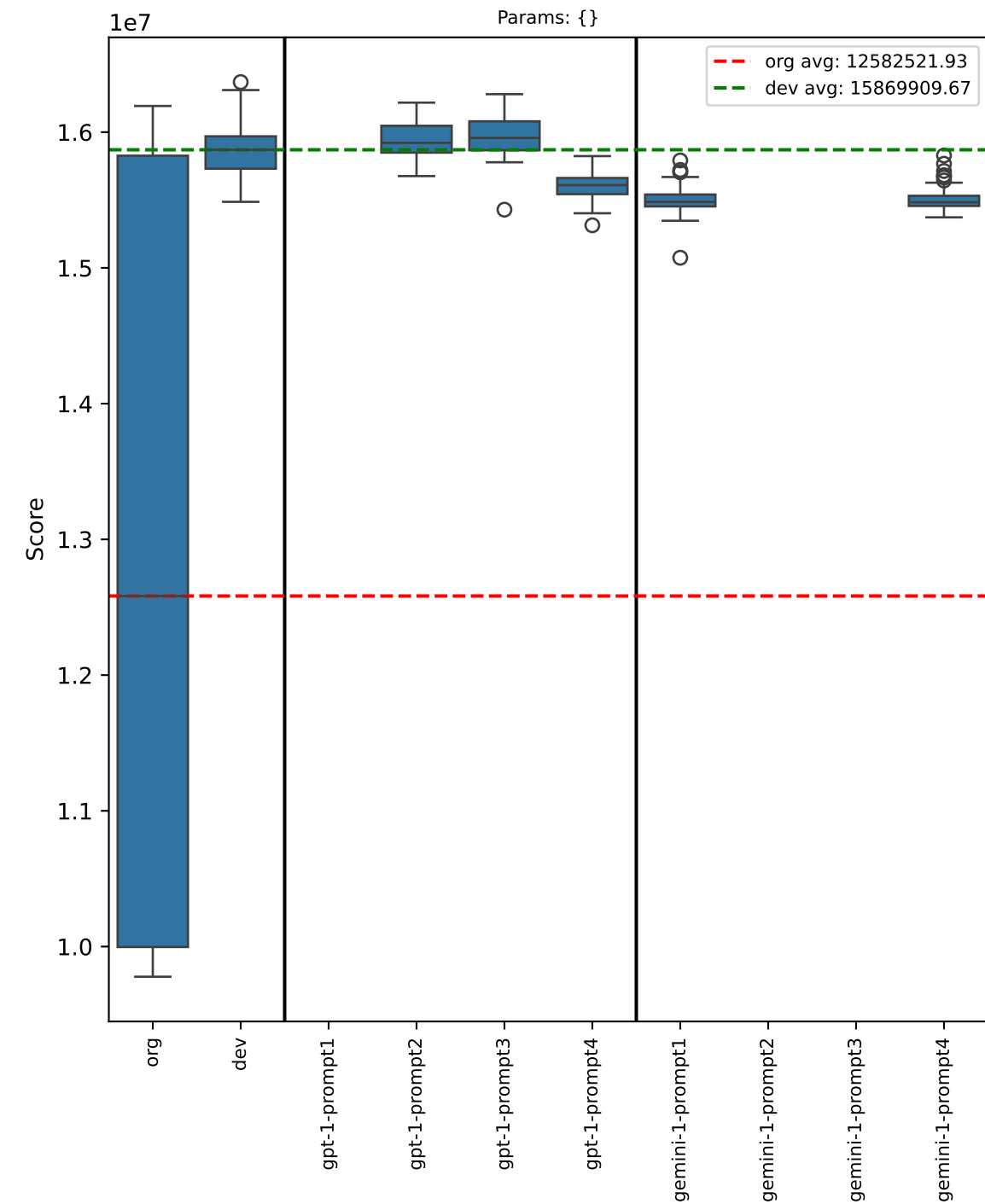




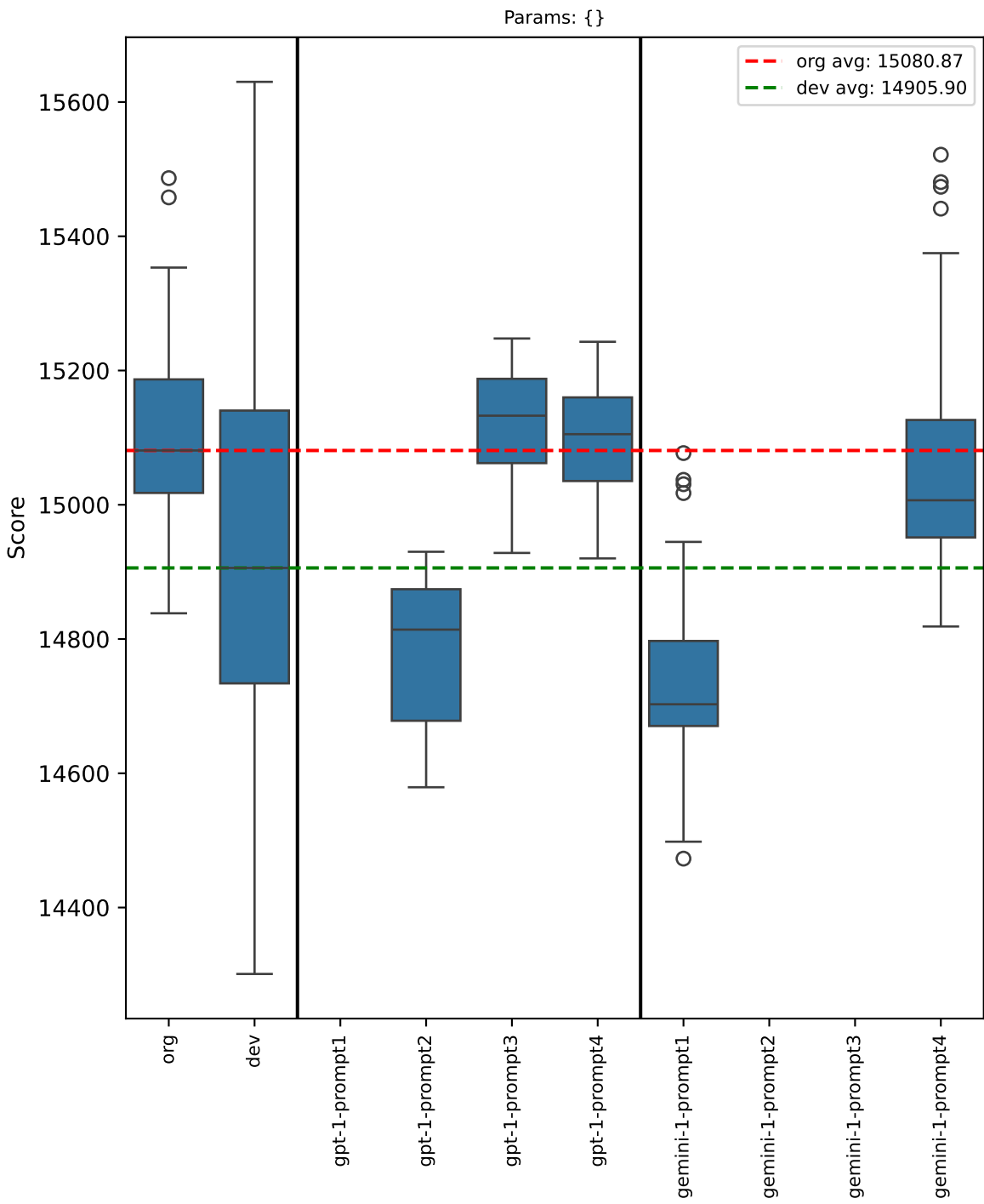


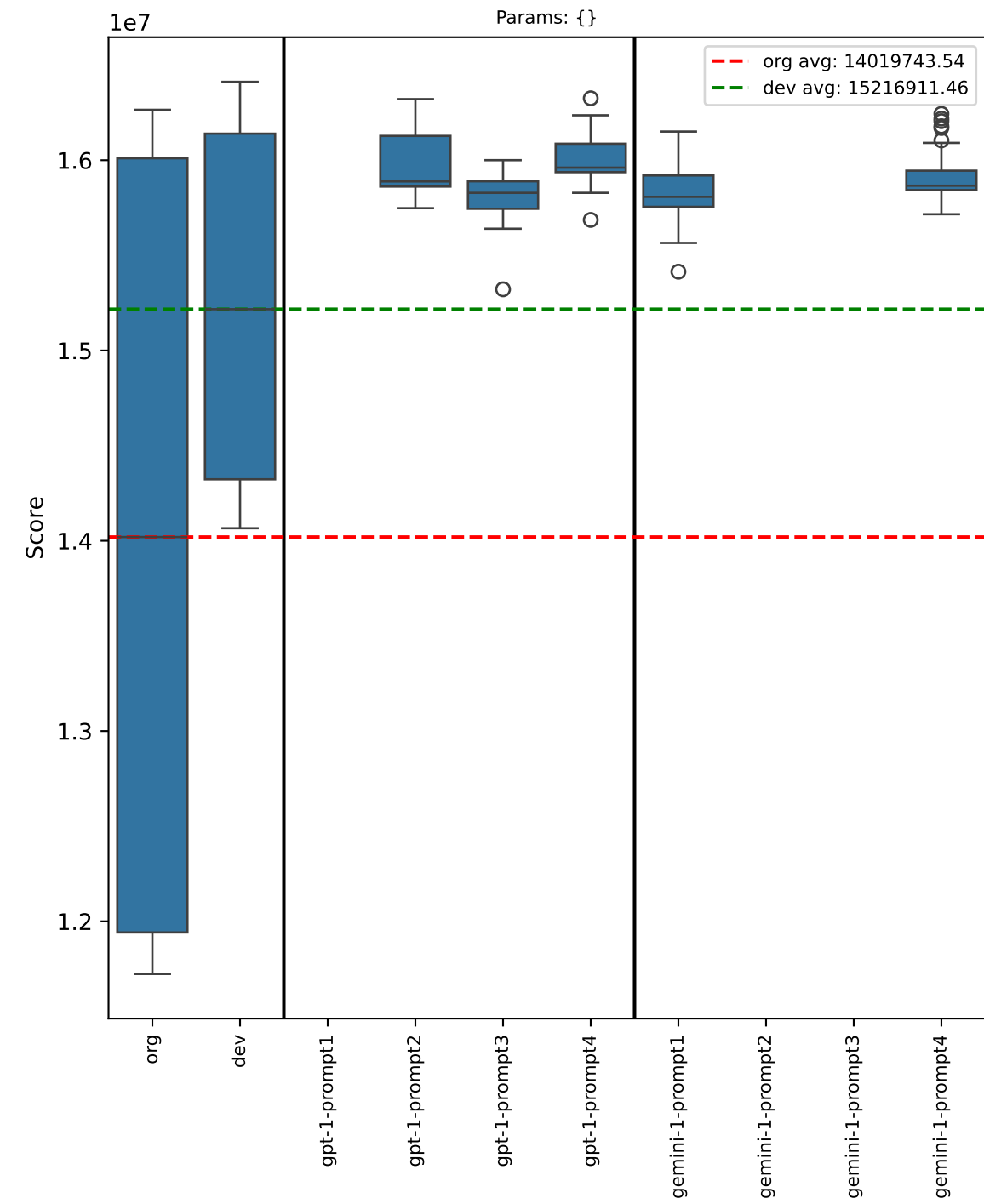




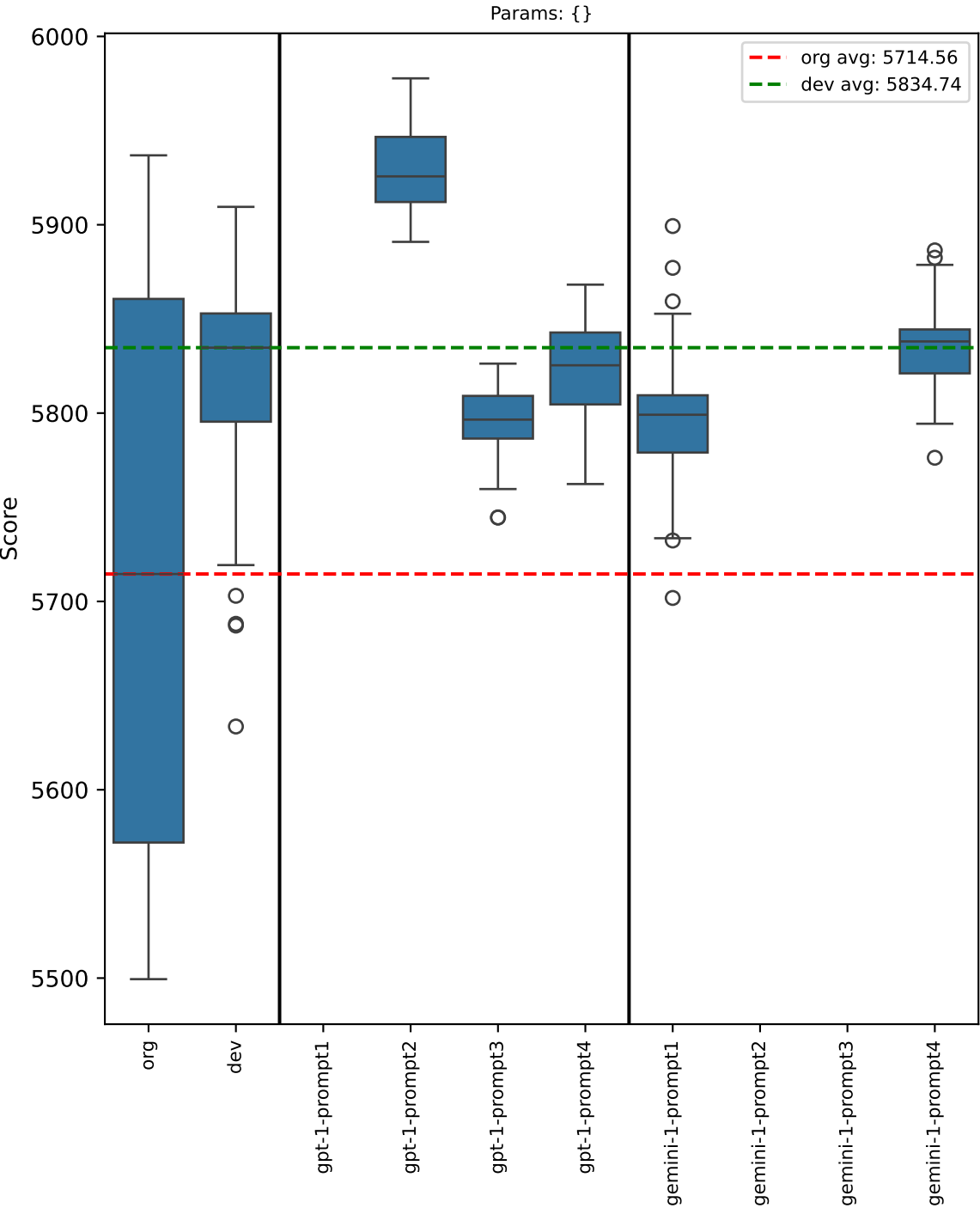


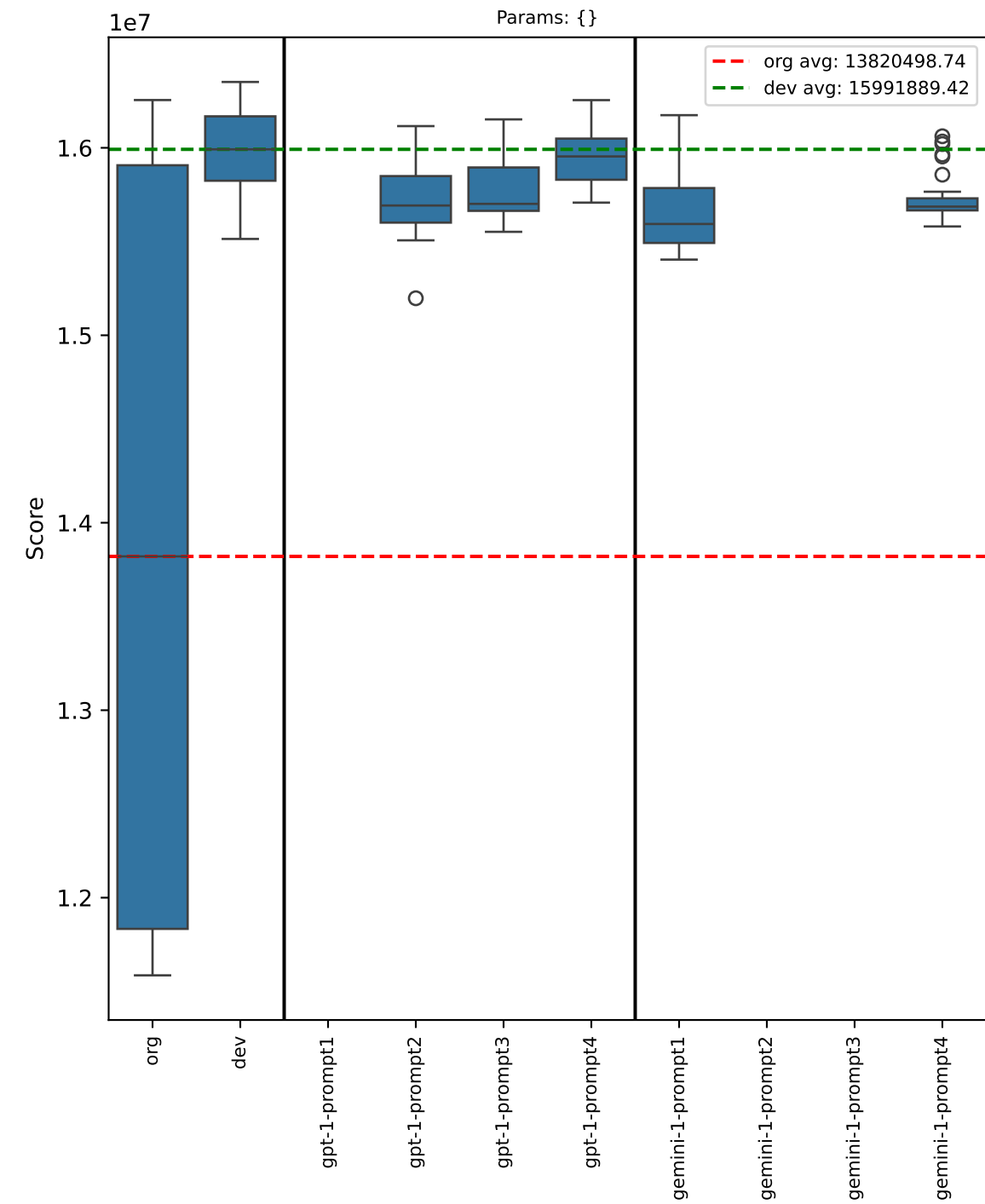
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeComplexMultipoint



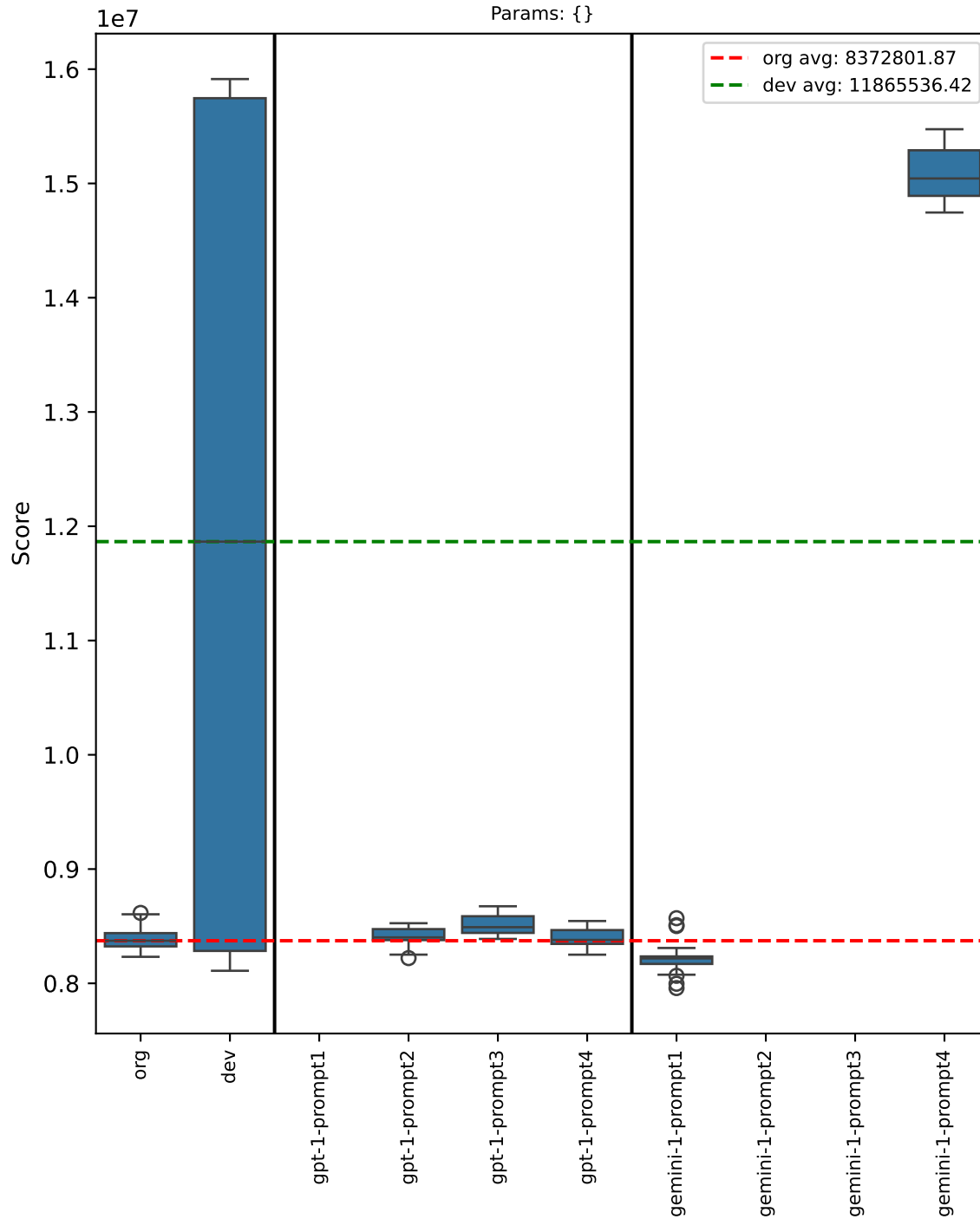


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeComplexPolygon

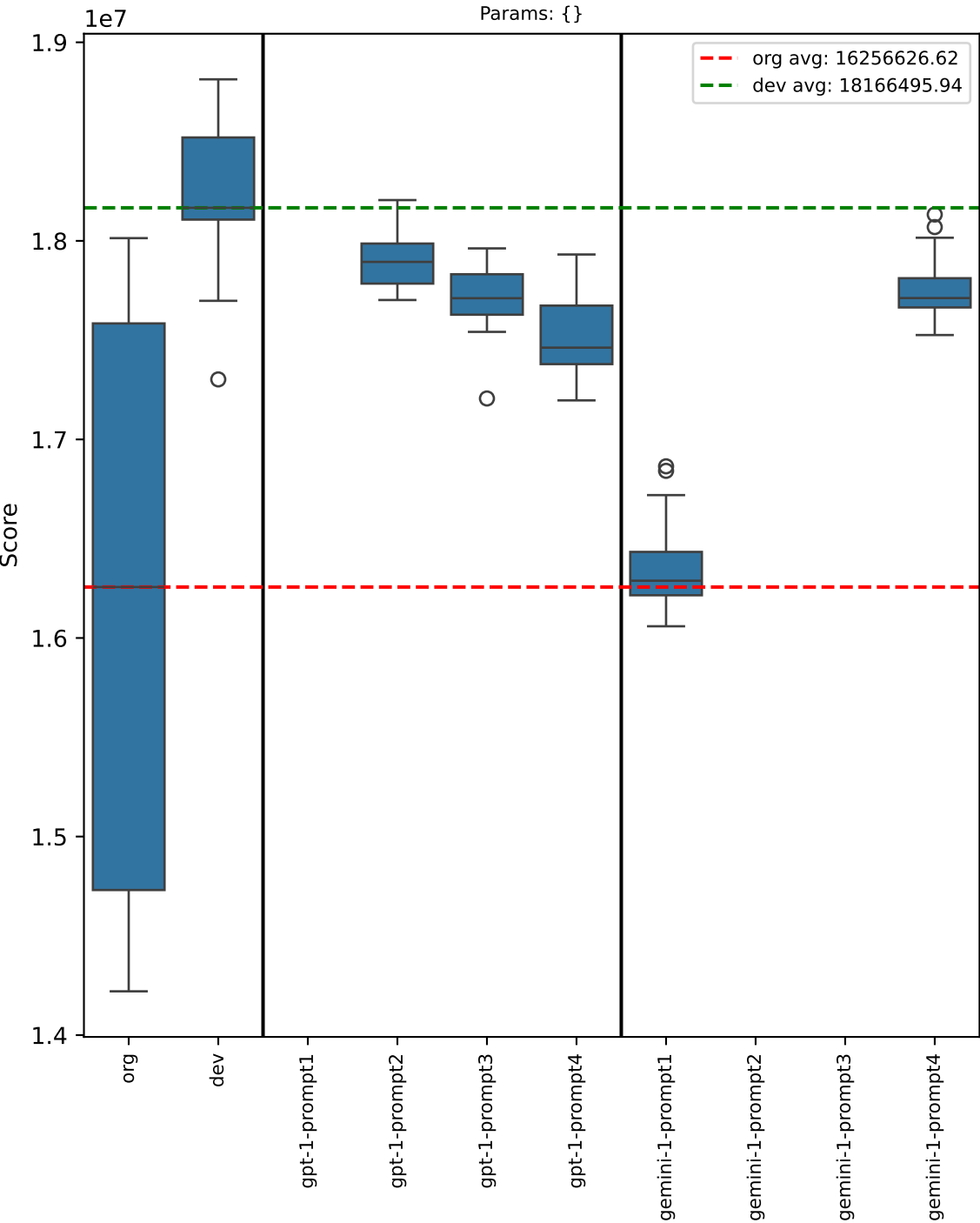


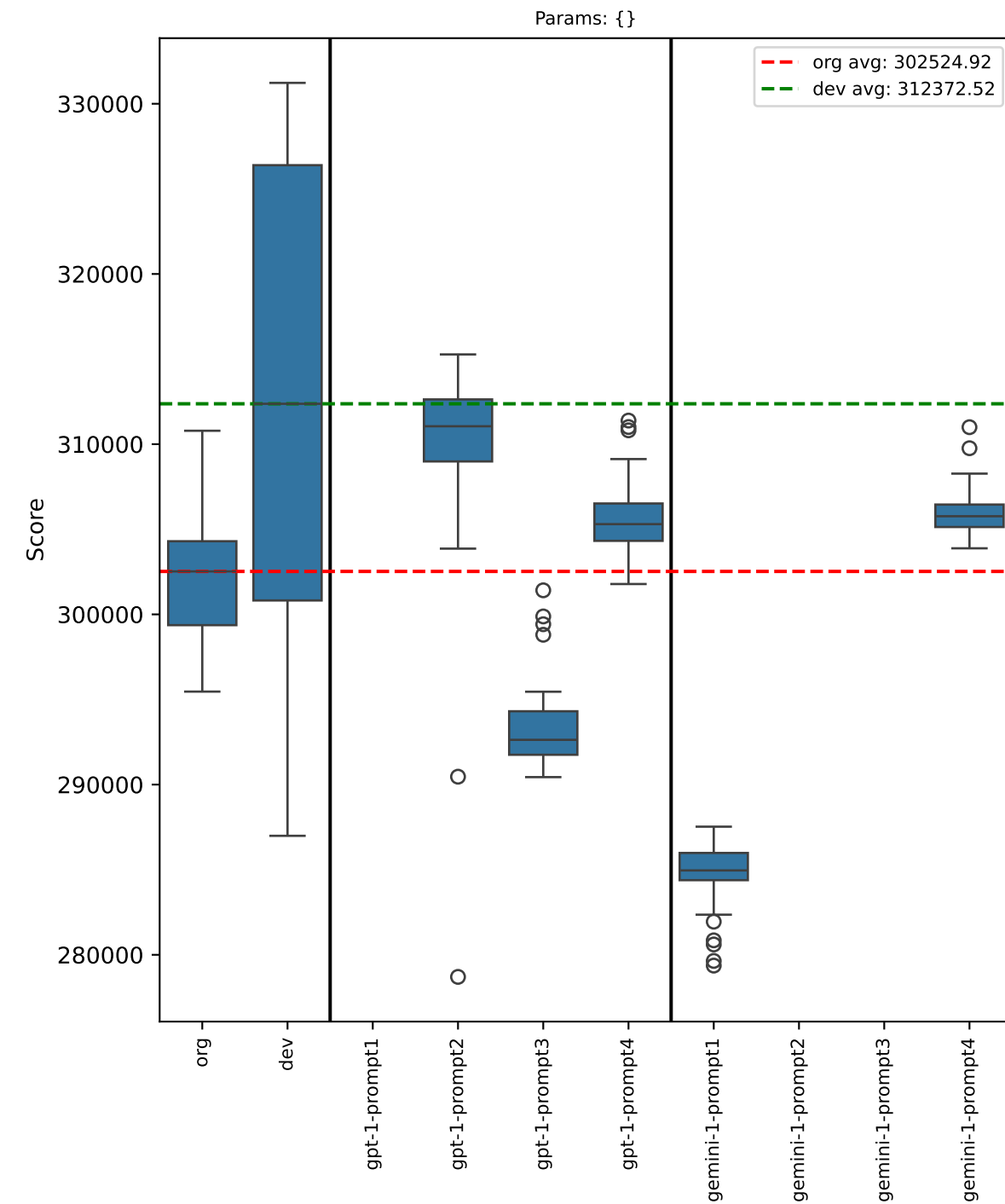


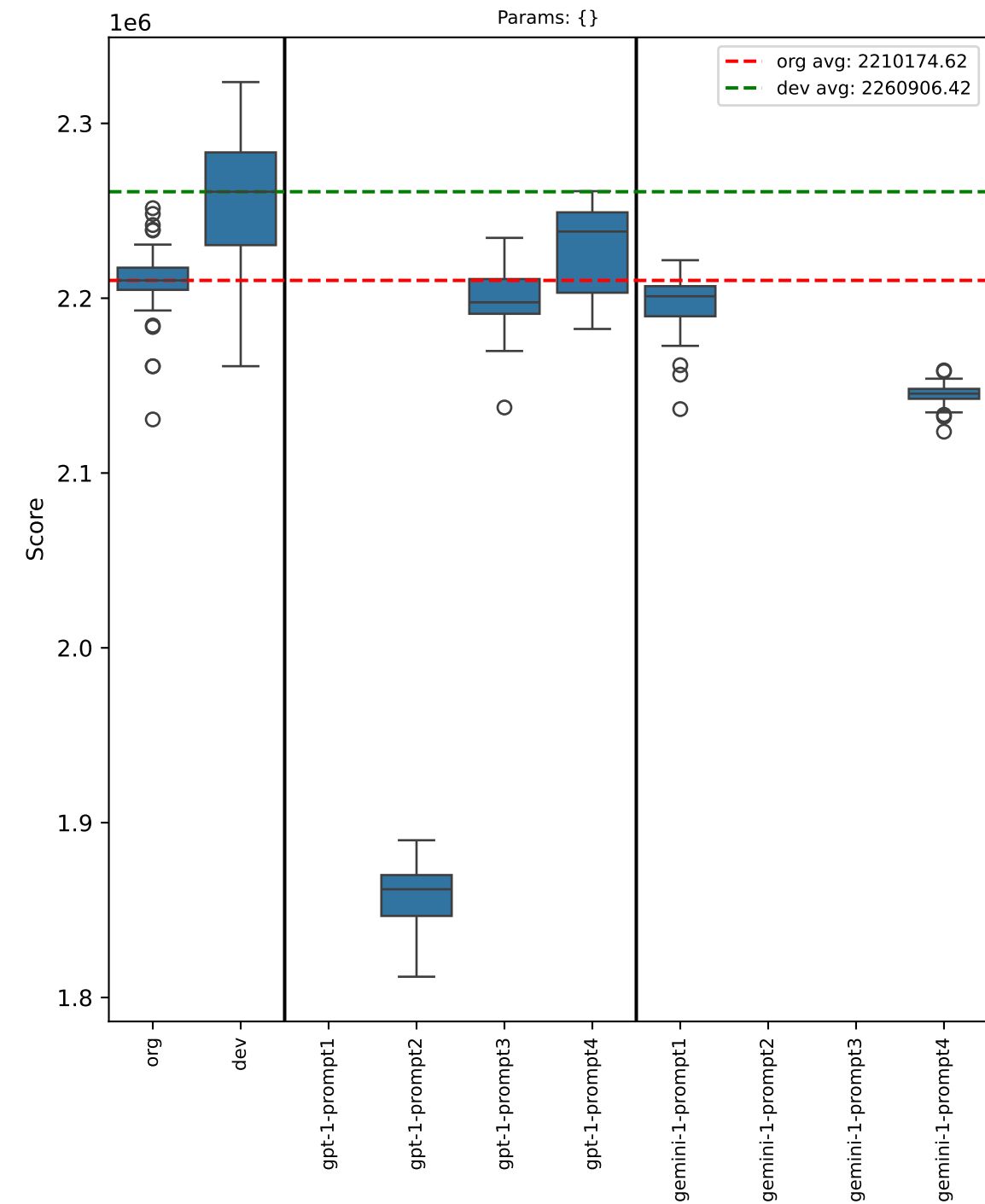
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializePoint



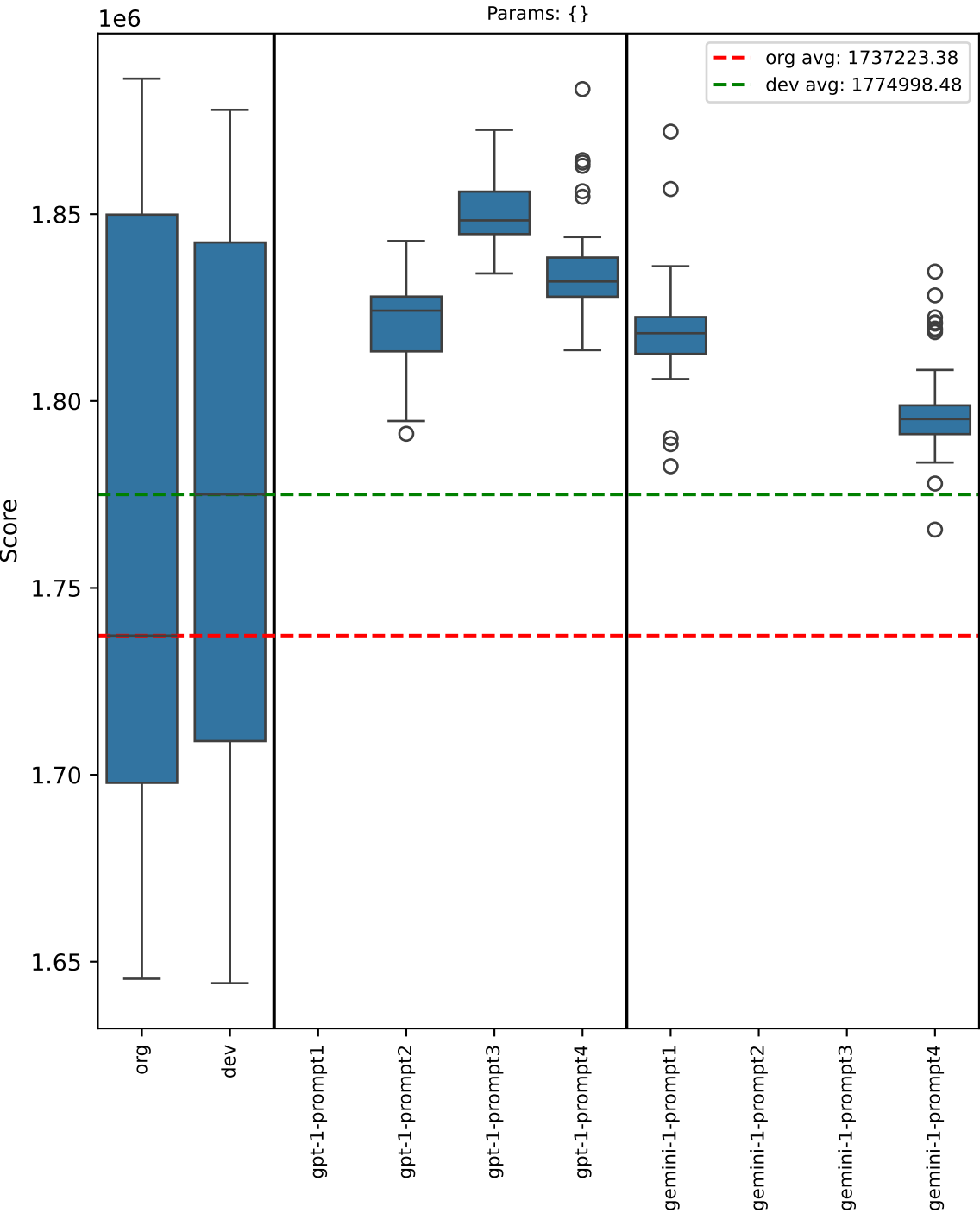
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializePointEnvelope

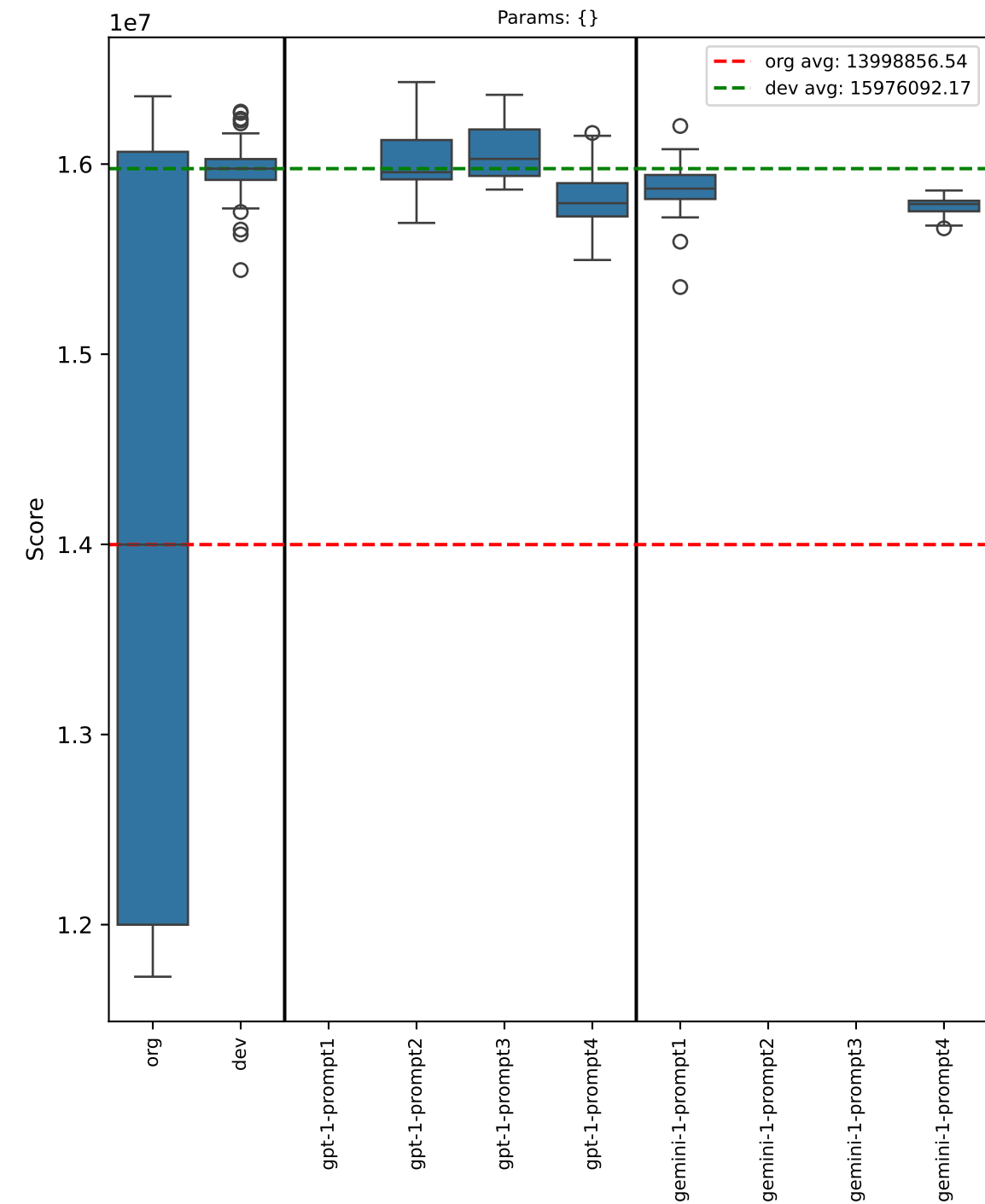


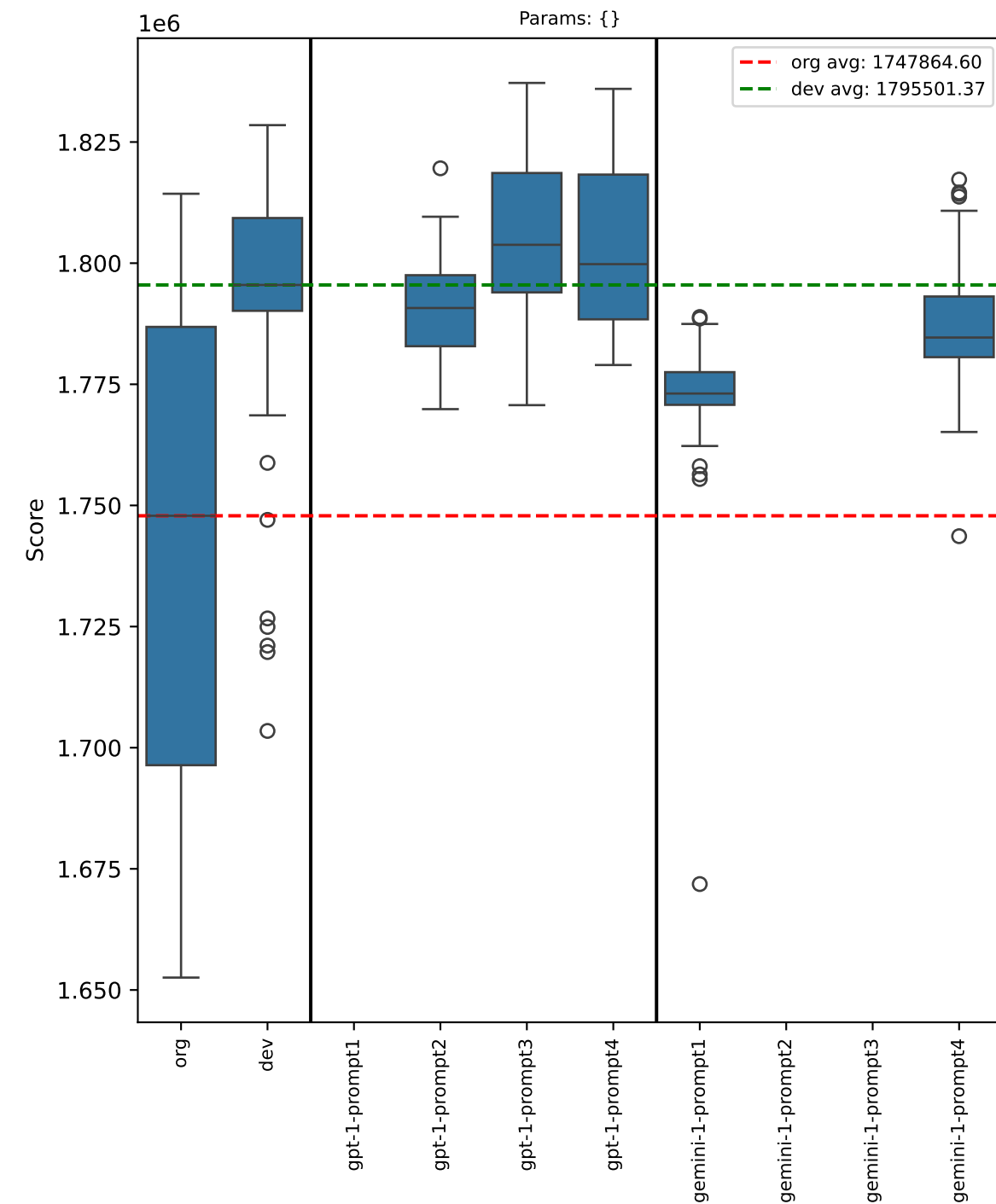


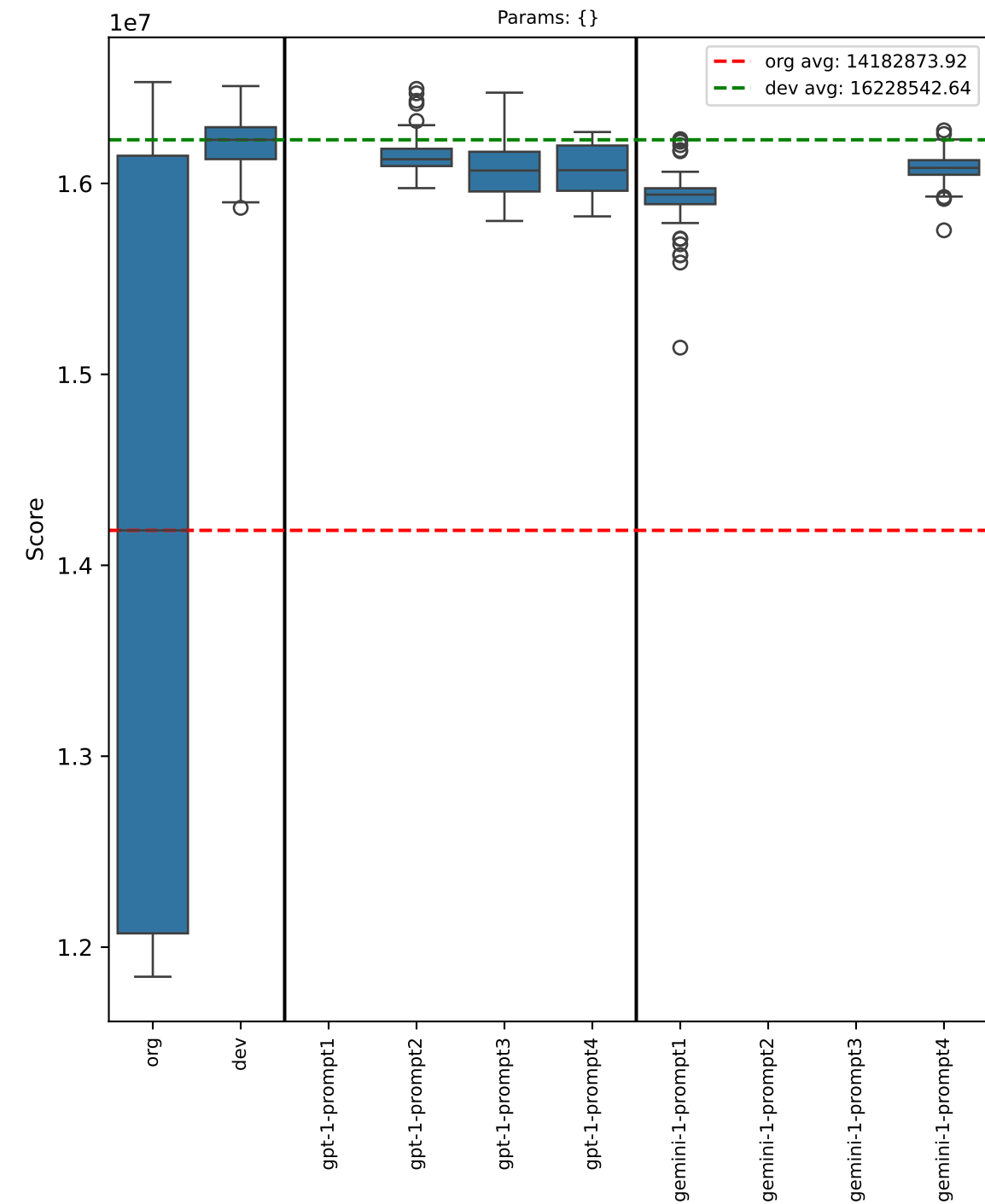


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeSimpleLineString

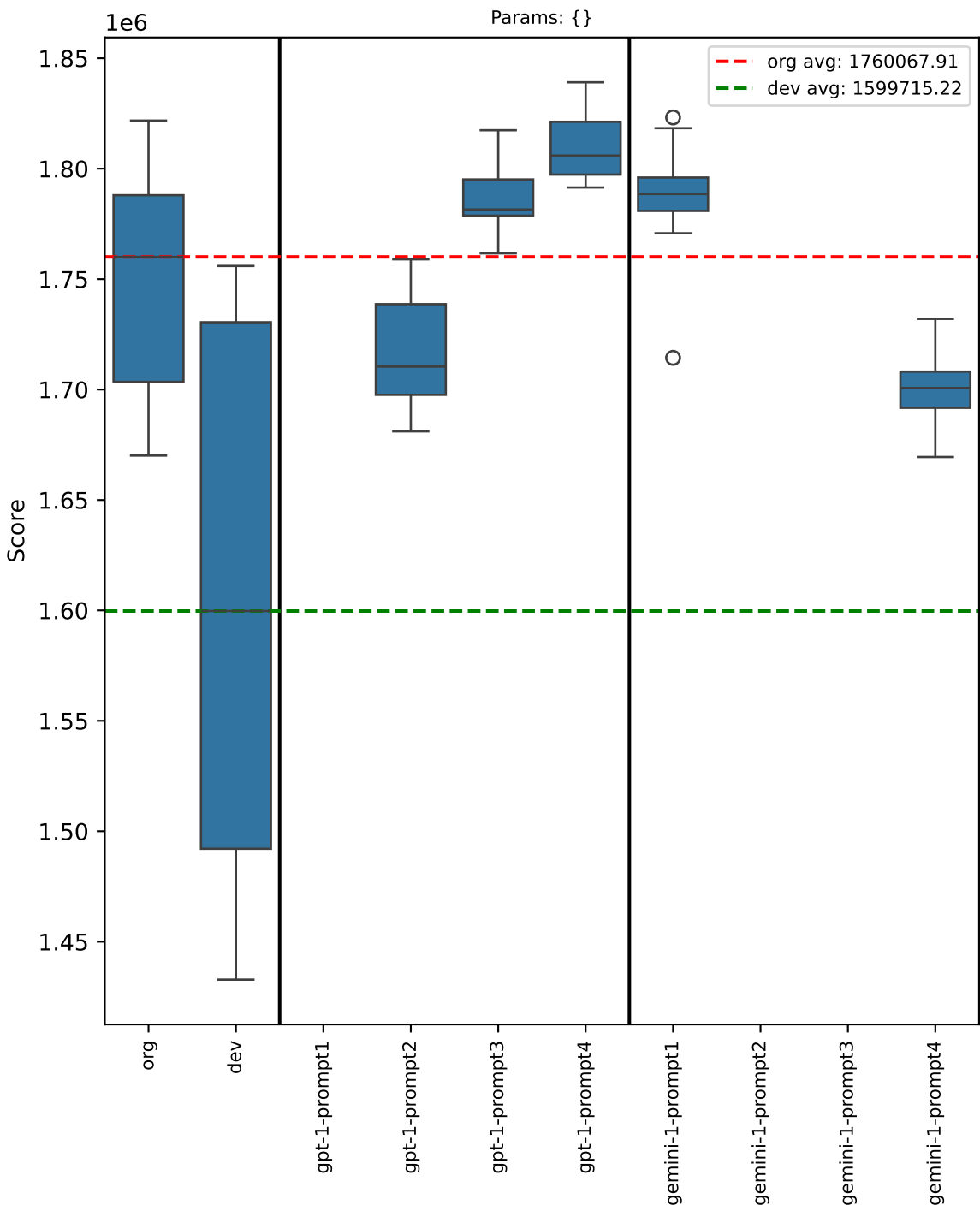


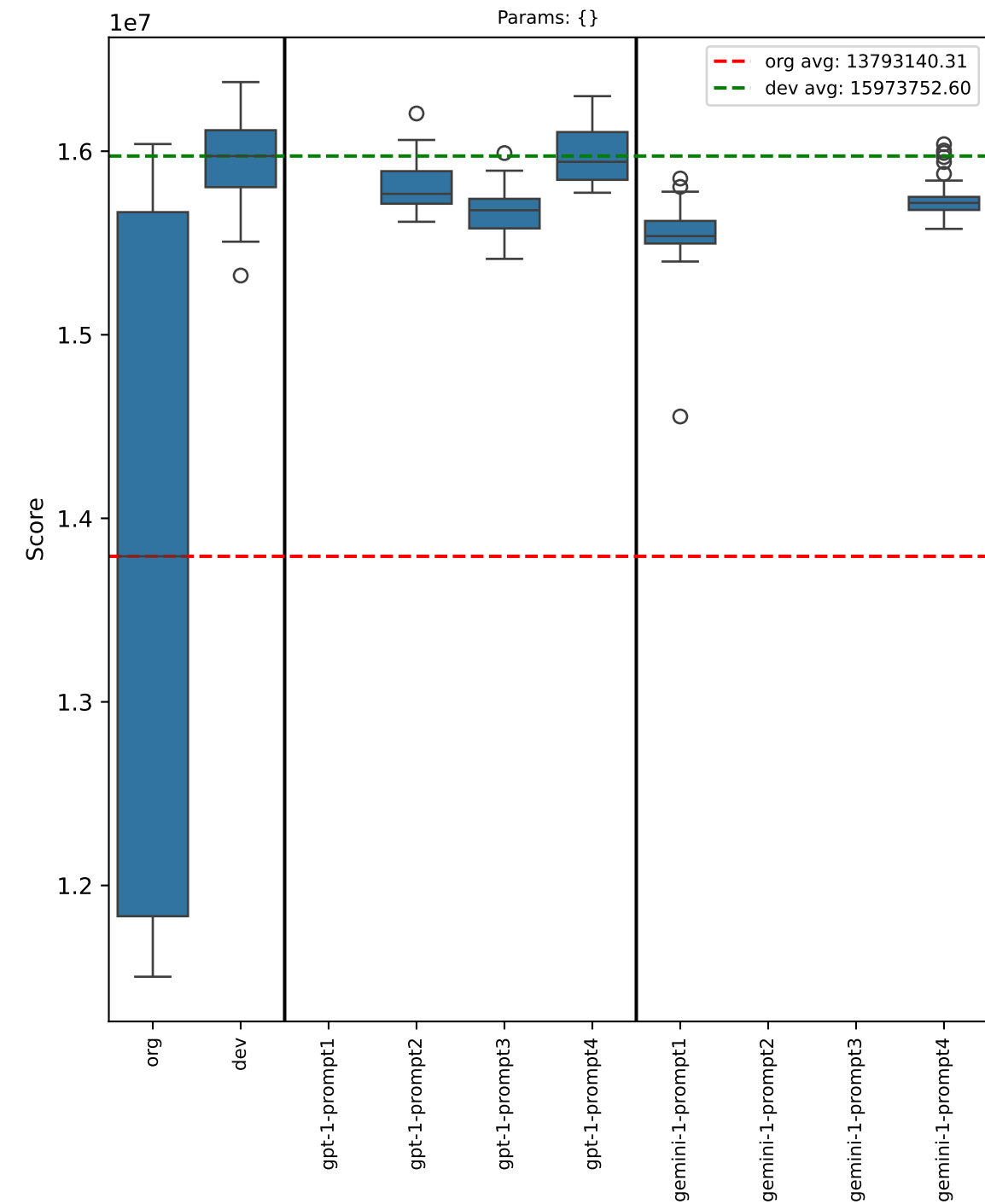




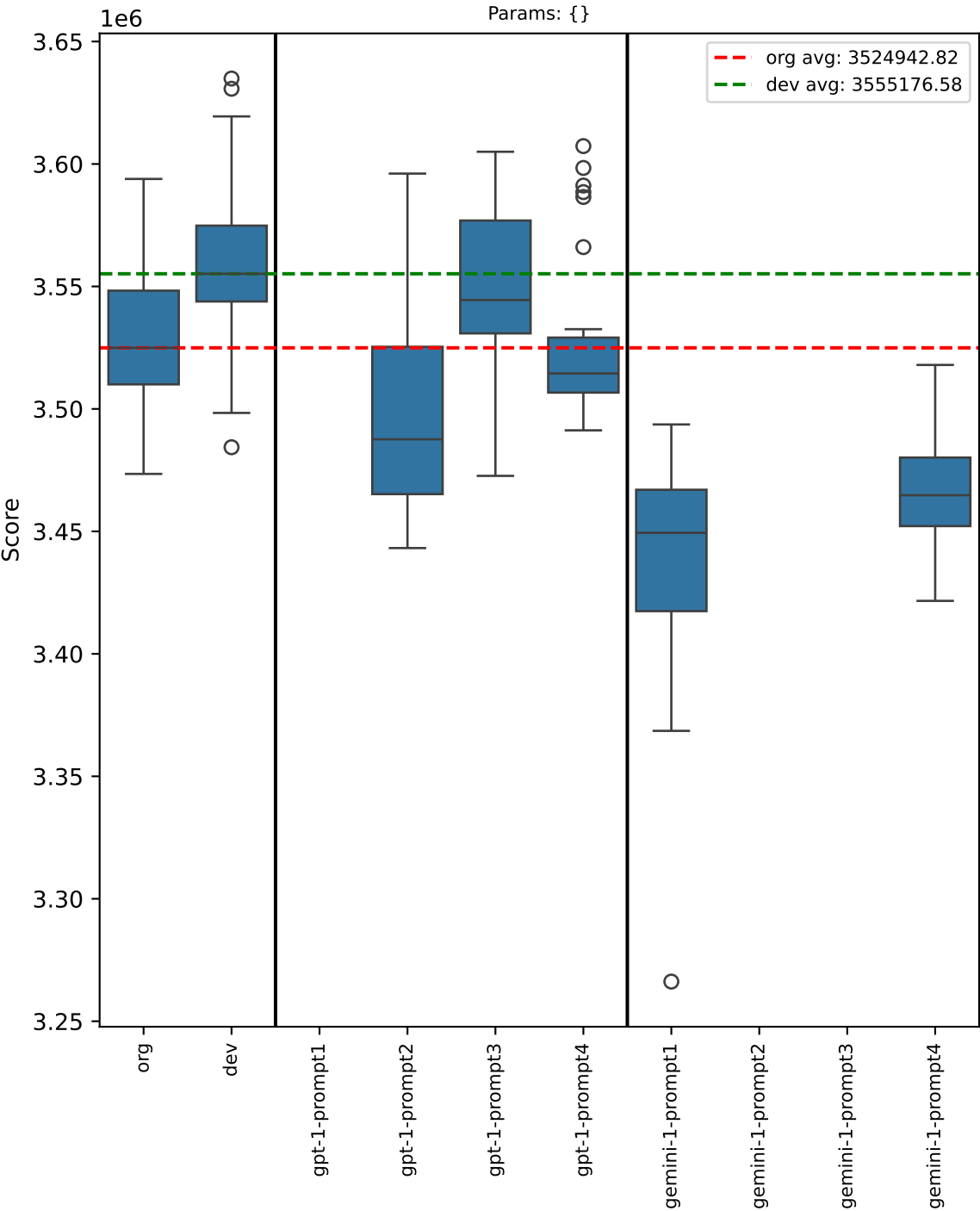


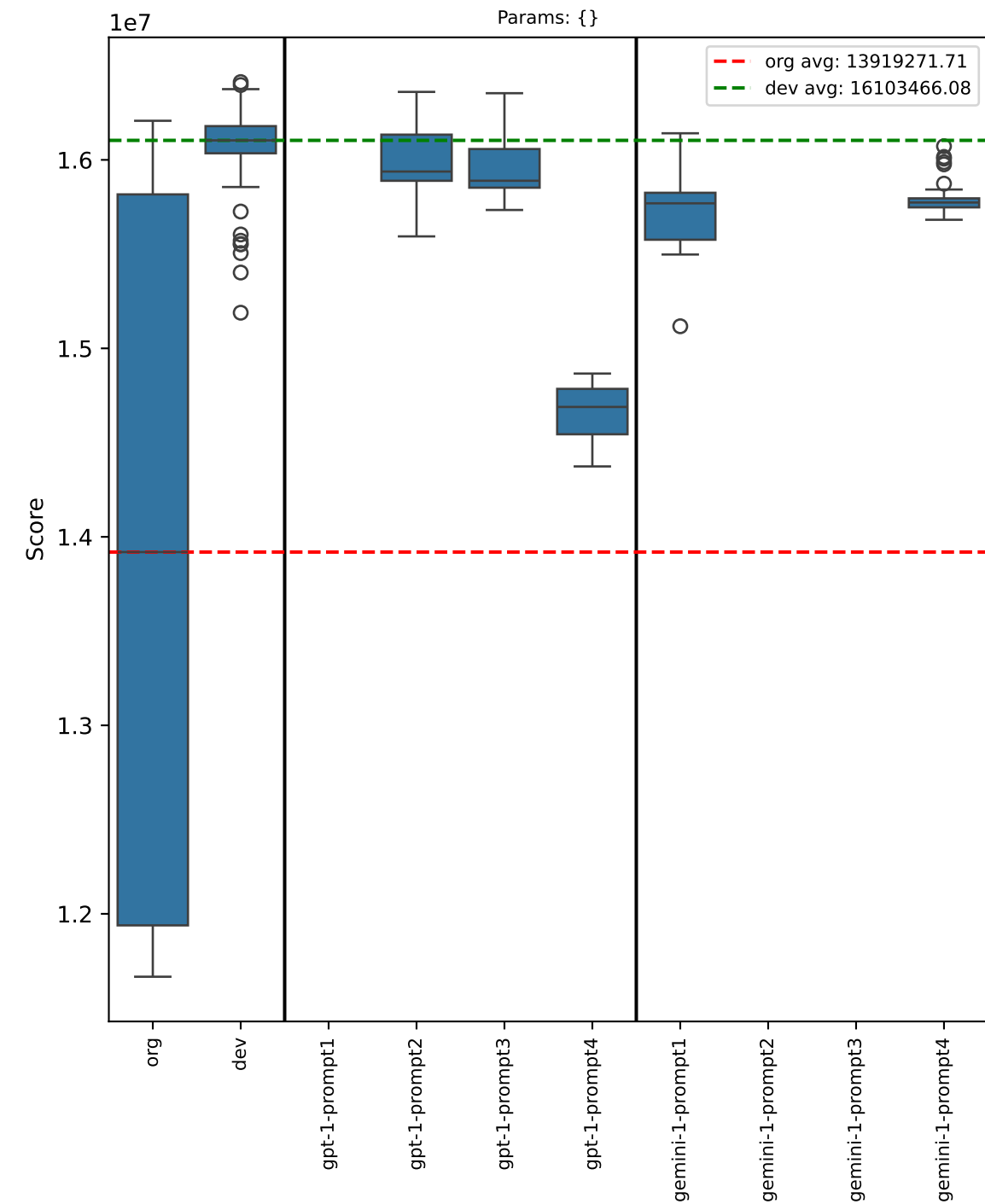
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeSimpleMultiPolygon



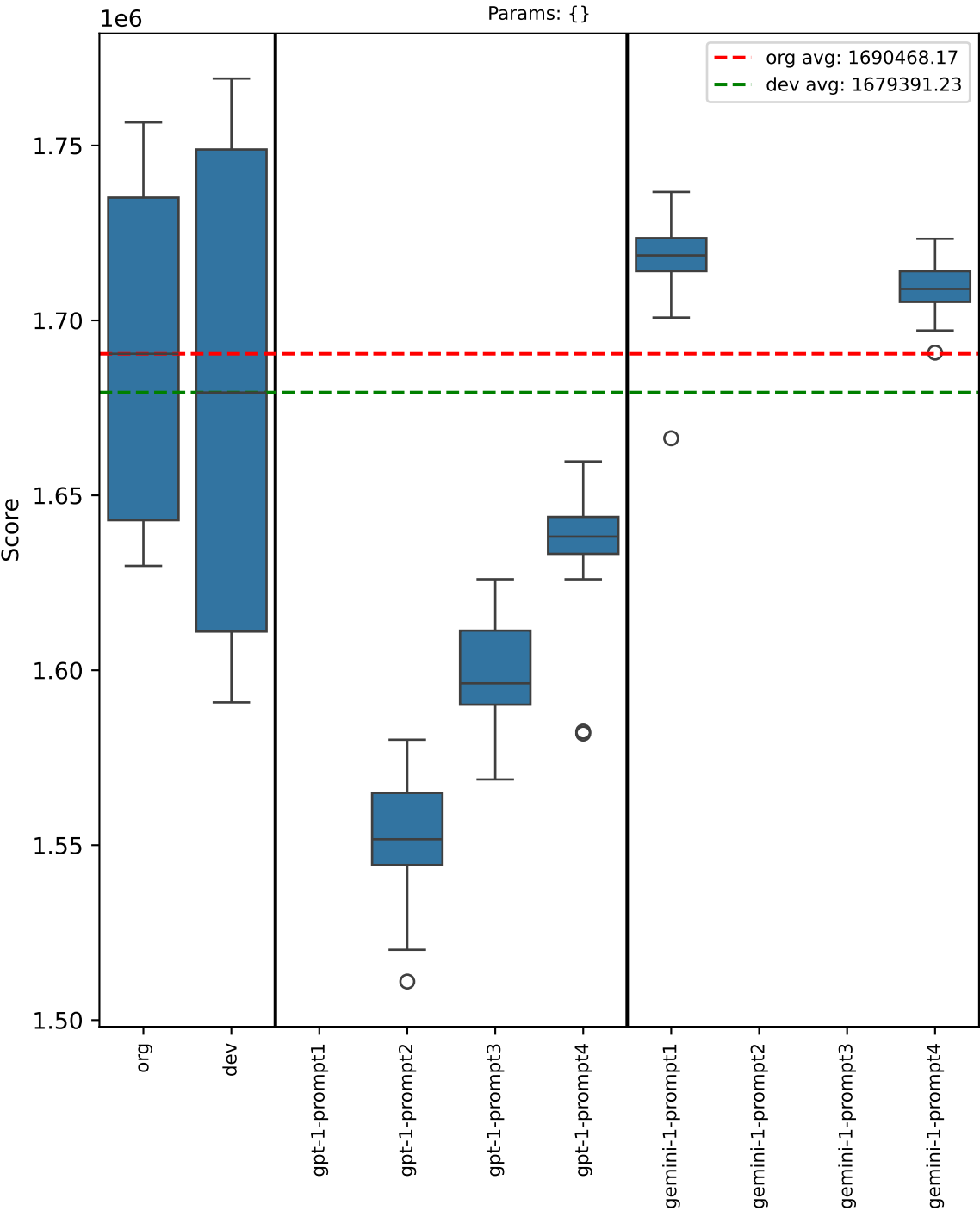


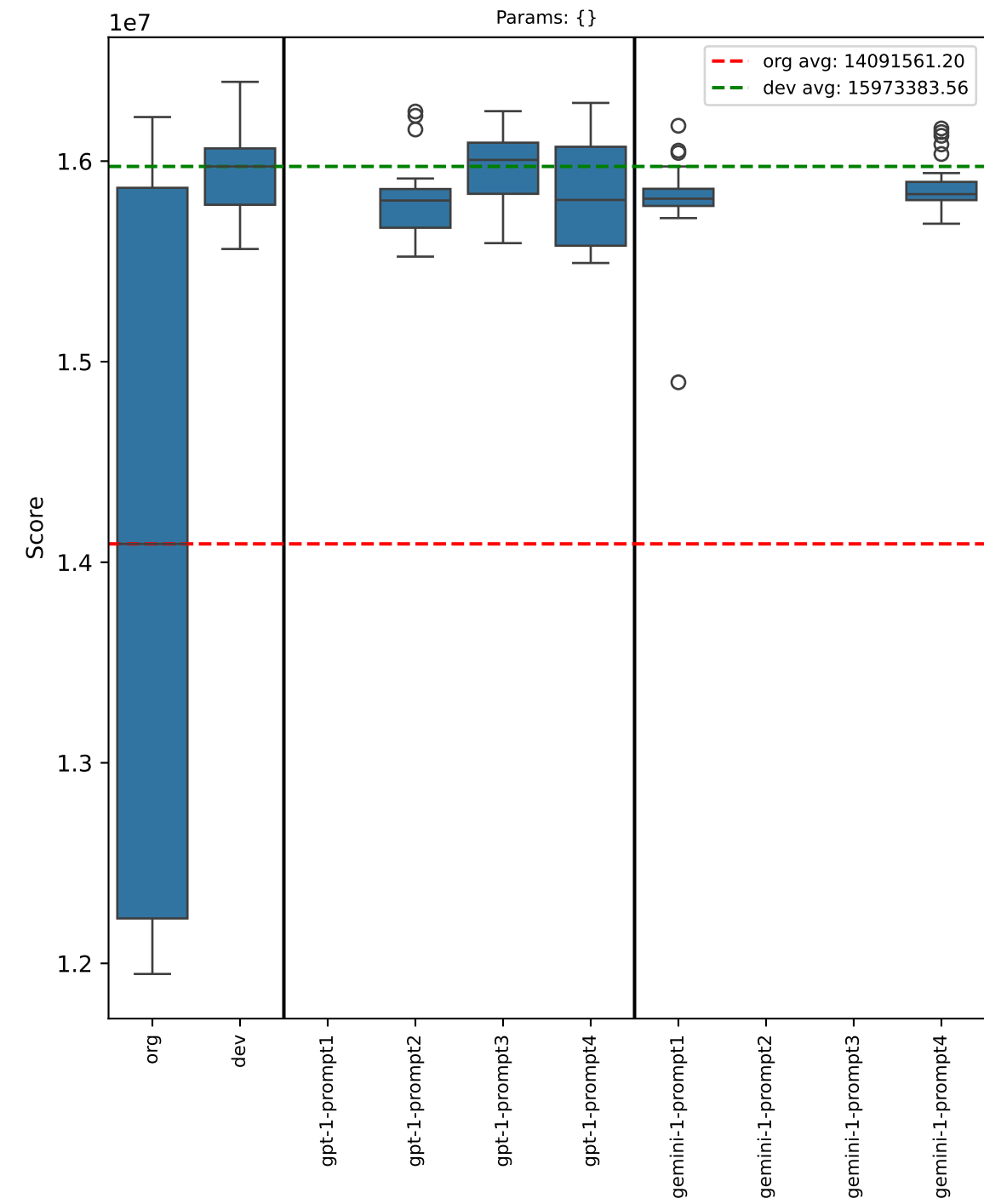
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeSimpleMultipoint

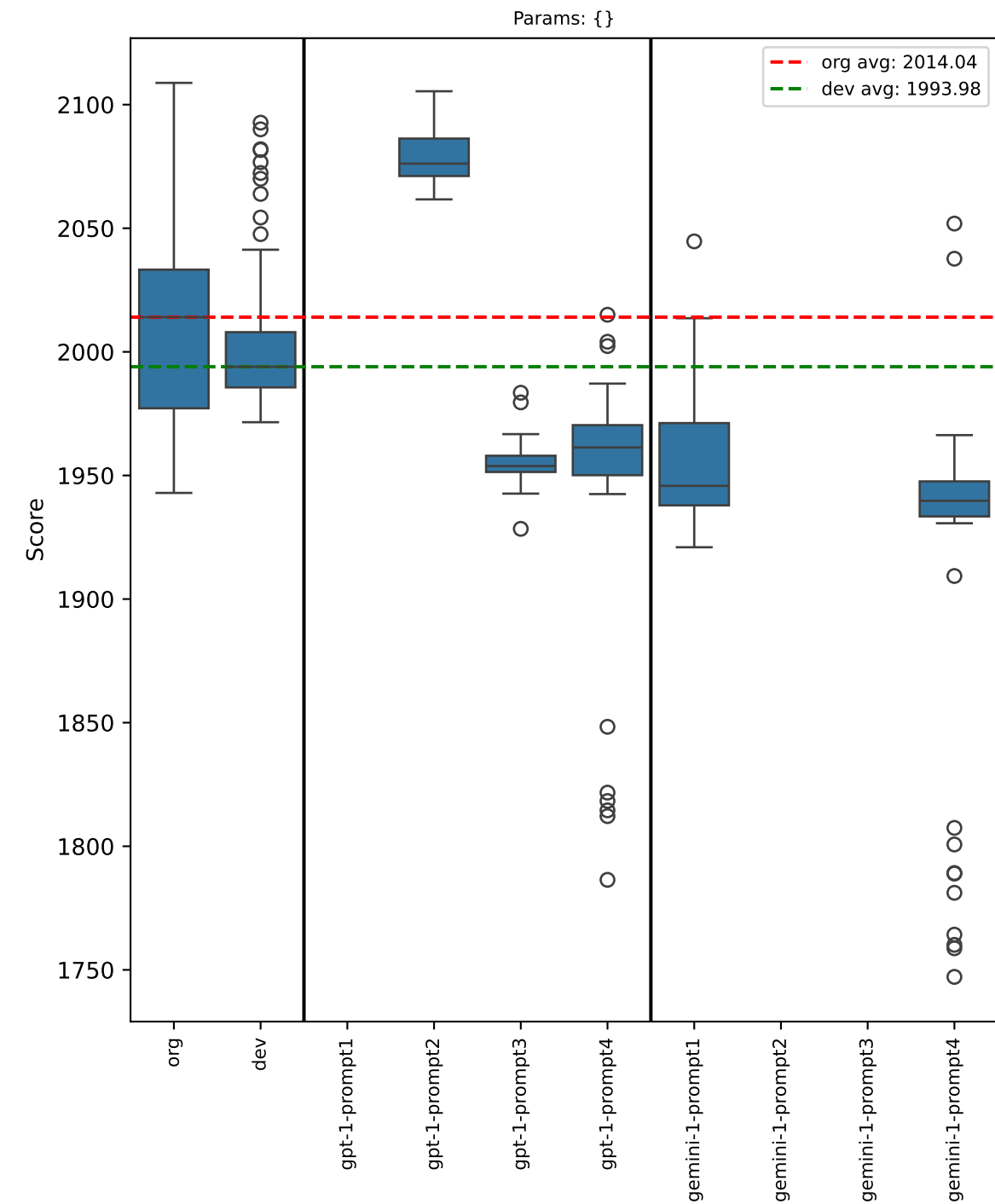




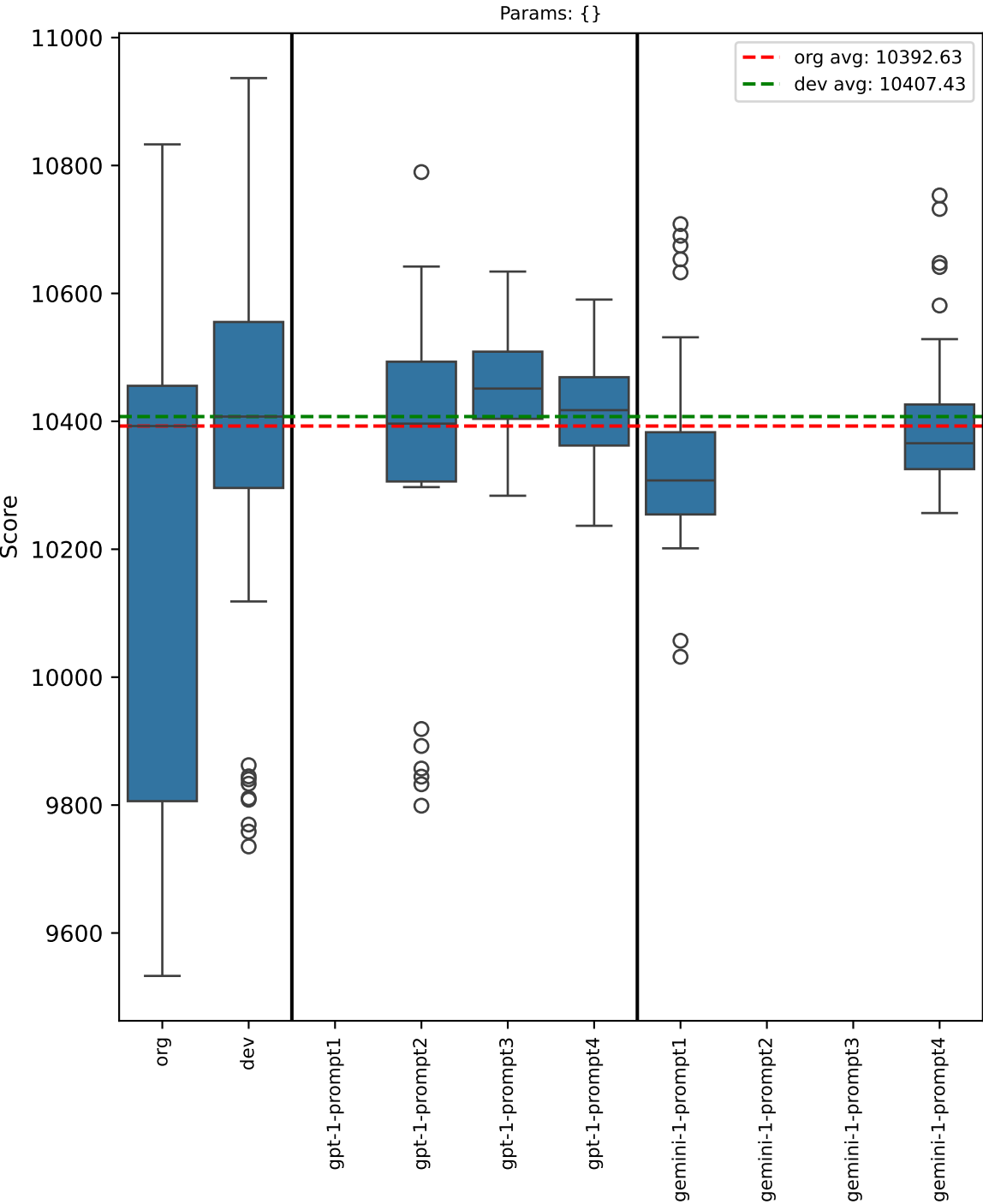
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.deserializeSimplePolygon

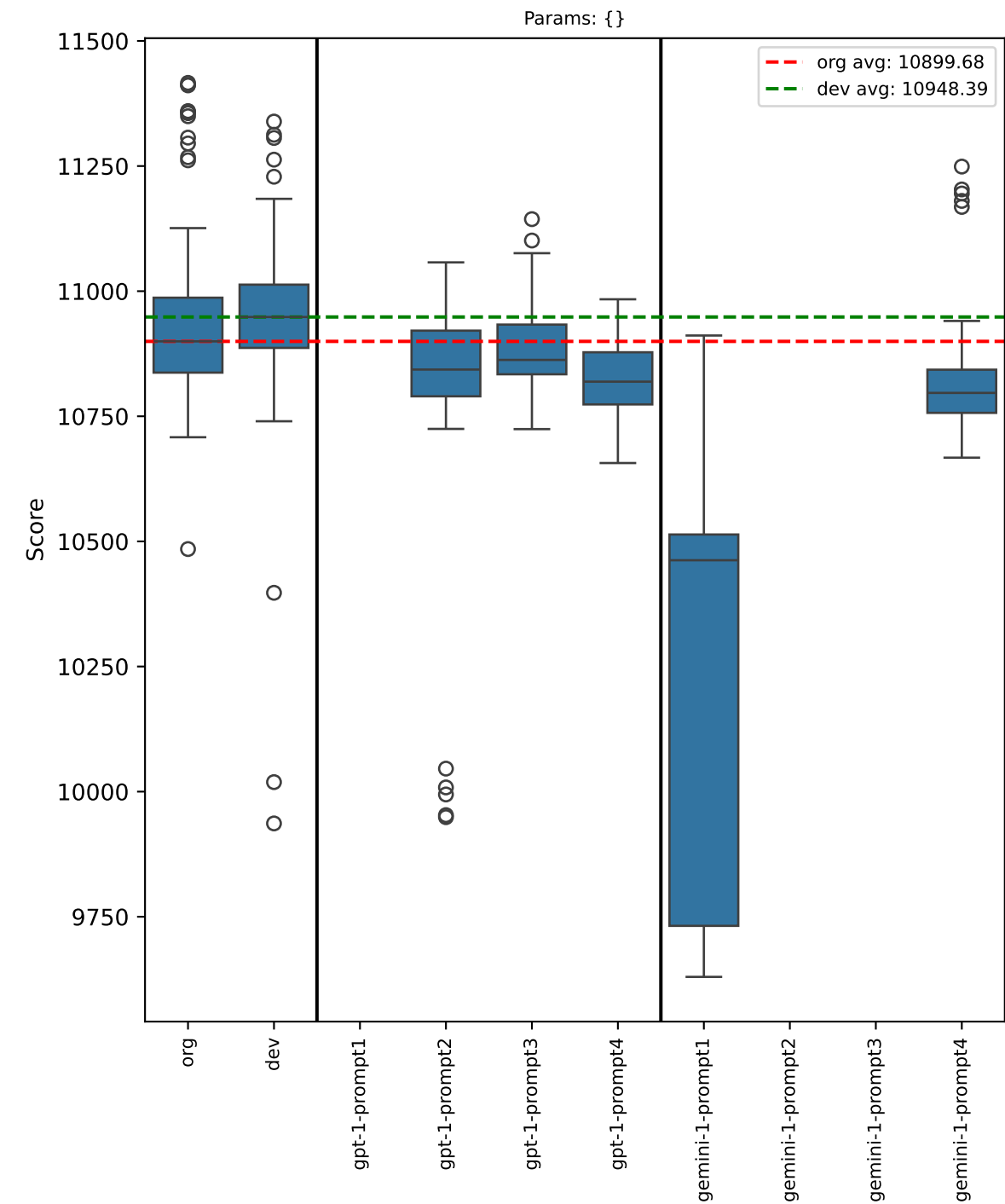




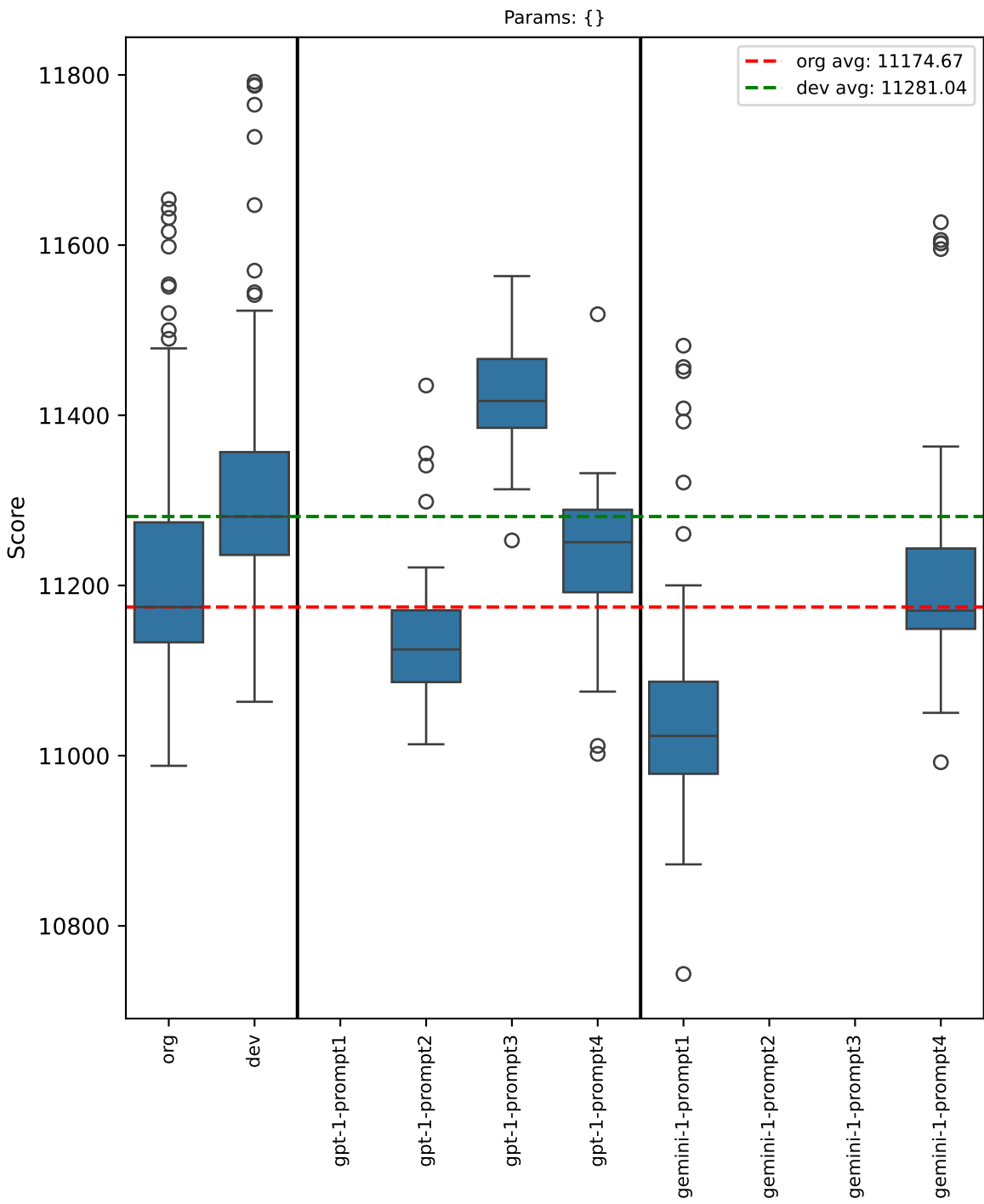


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeComplexLineString

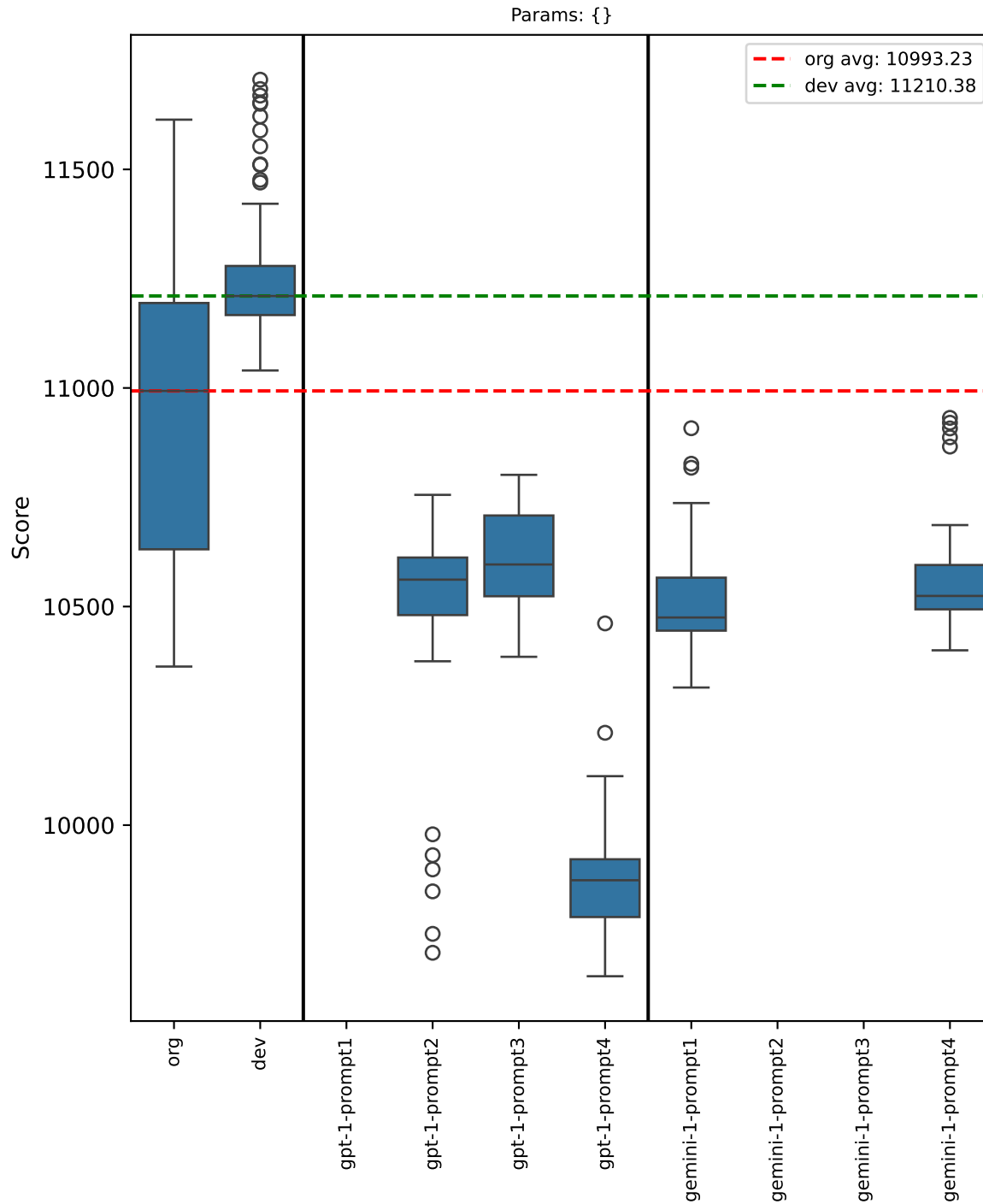




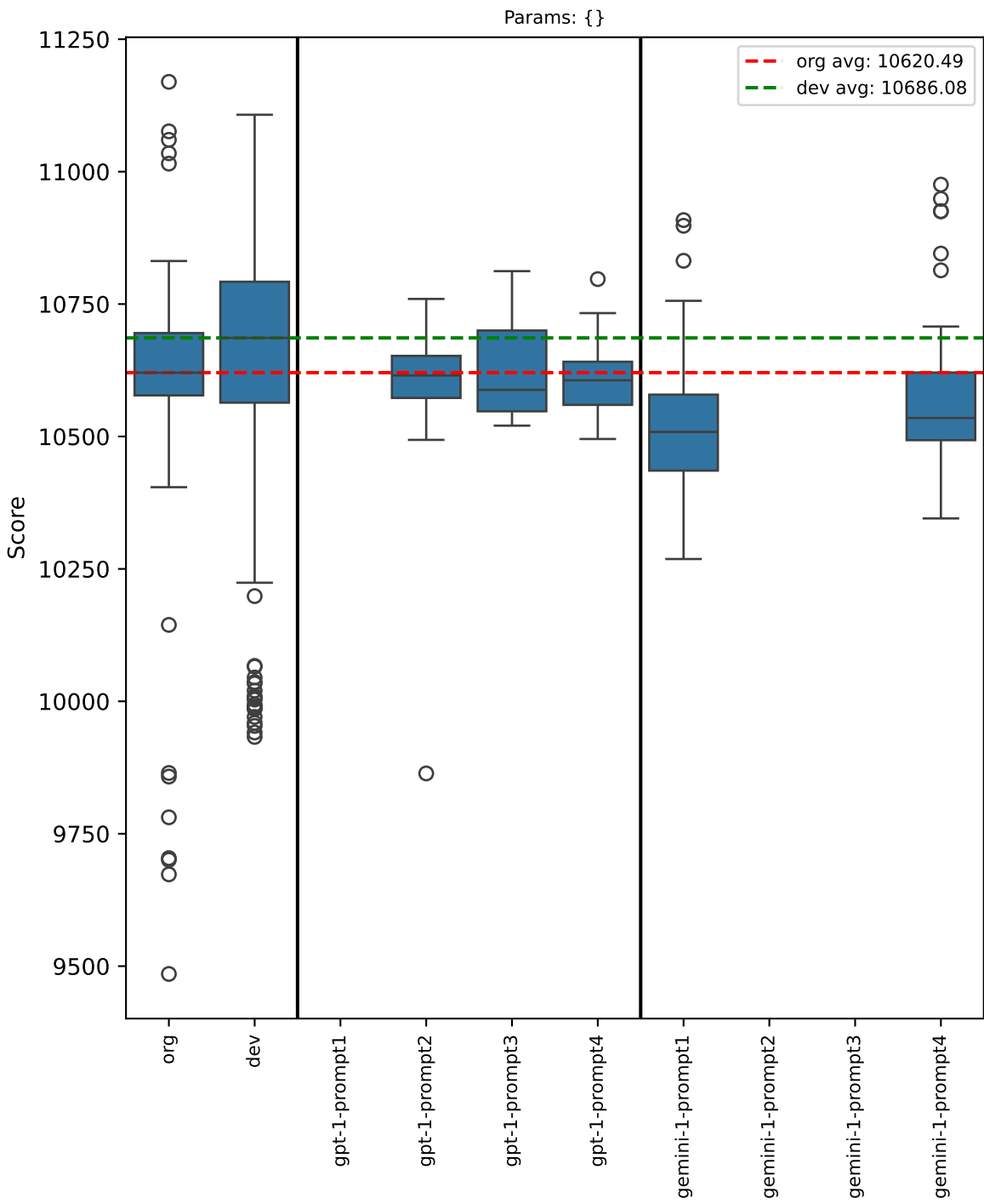
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeComplexMultiPolygon



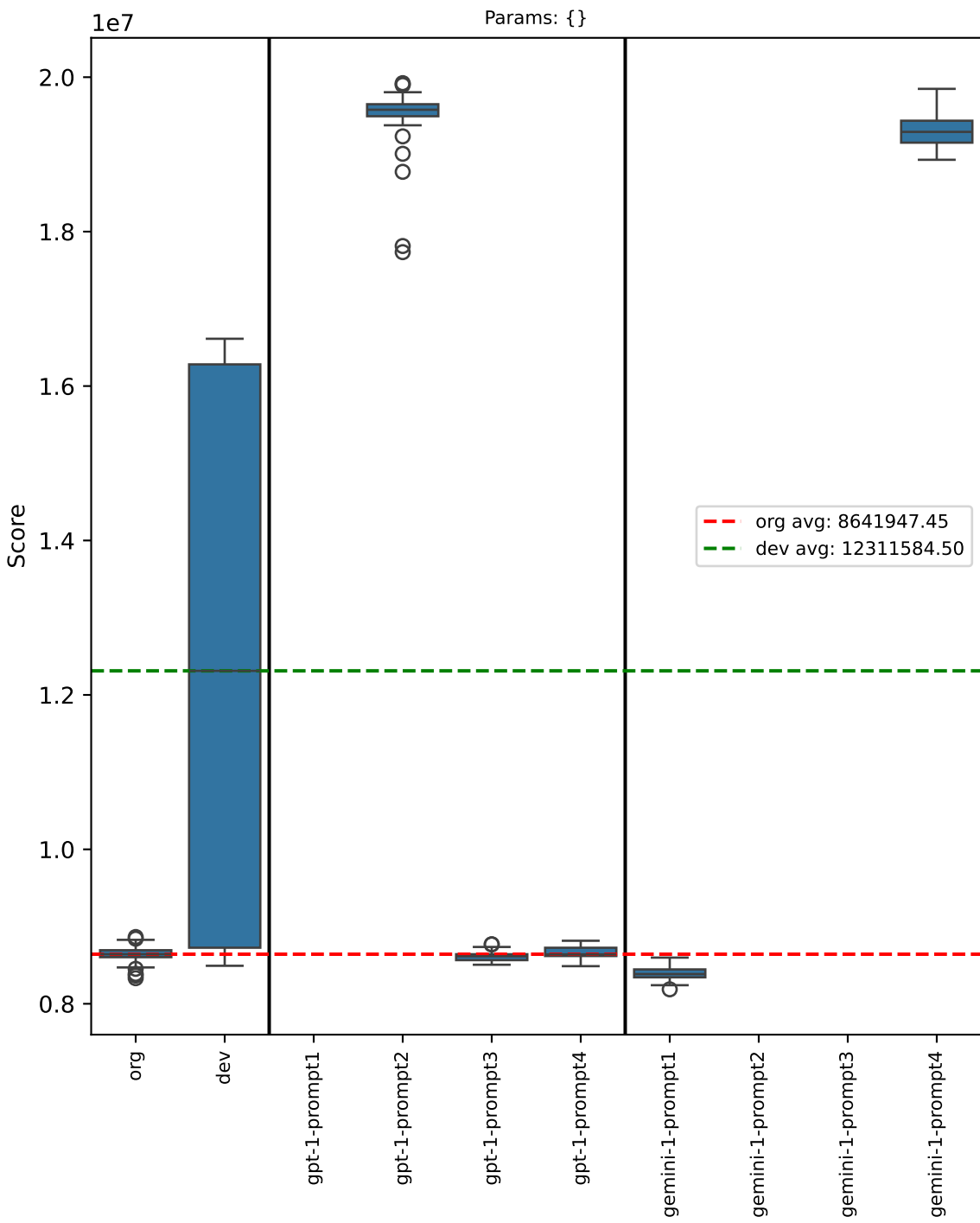
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeComplexMultipoint

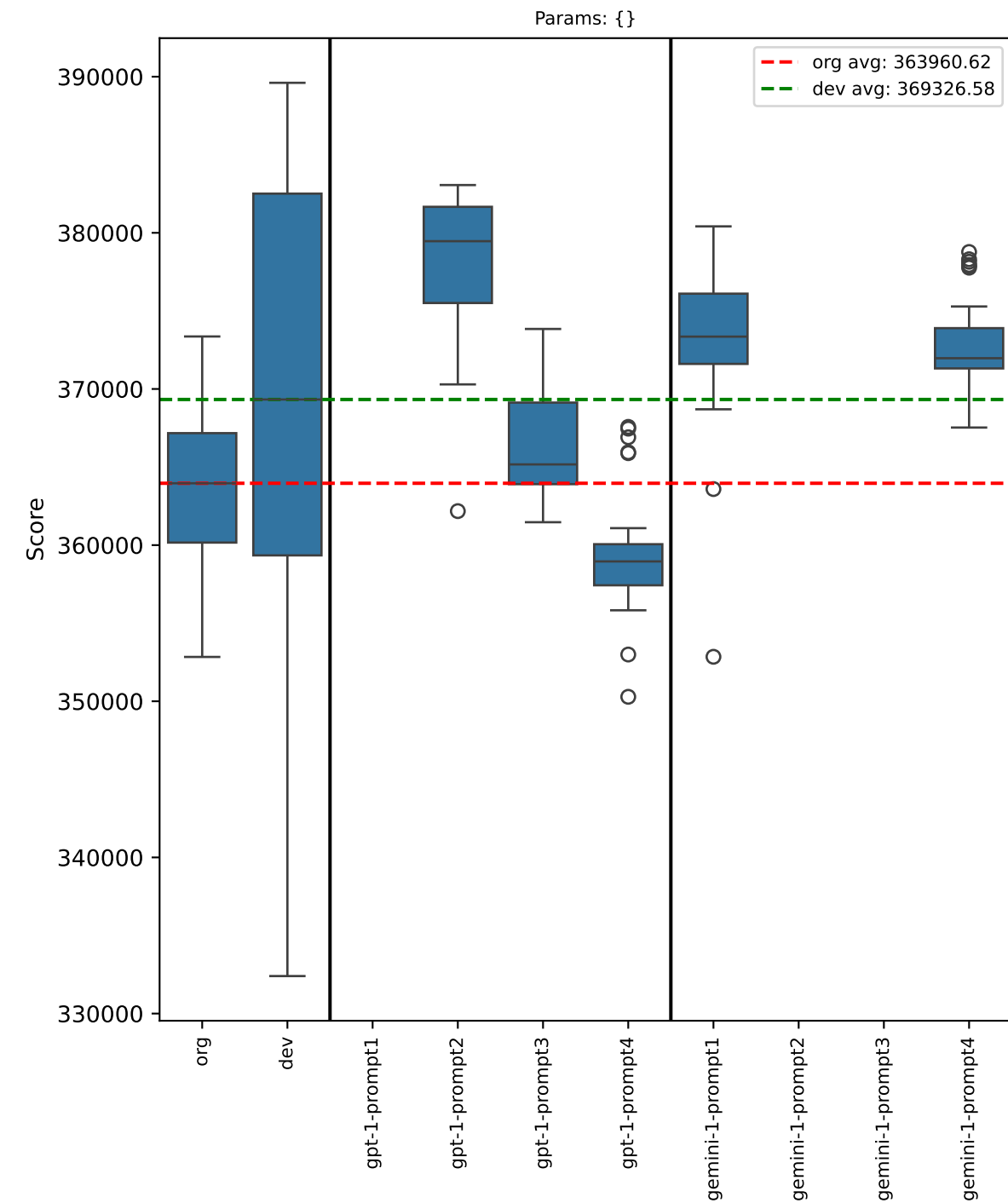


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeComplexPolygon

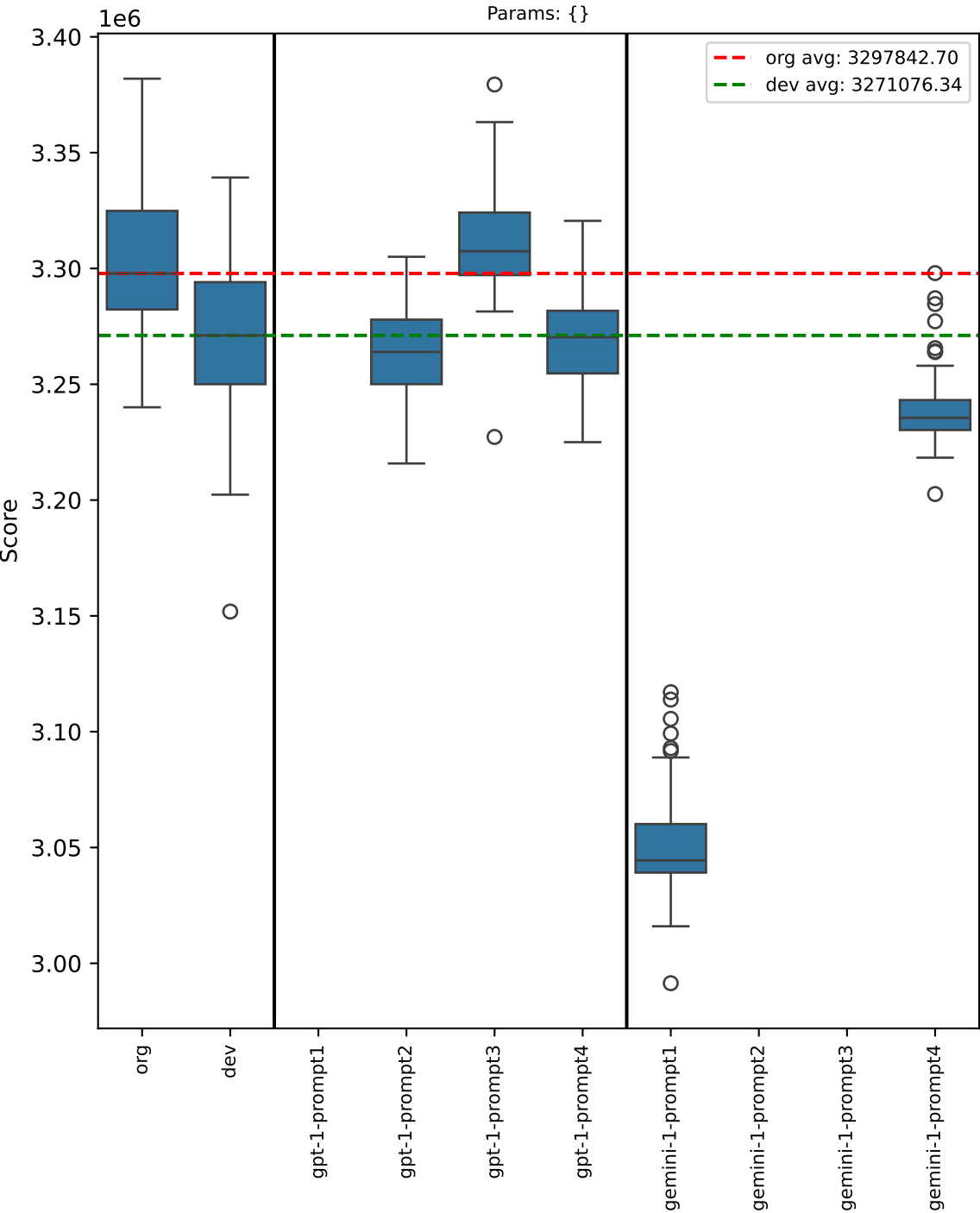


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializePoint

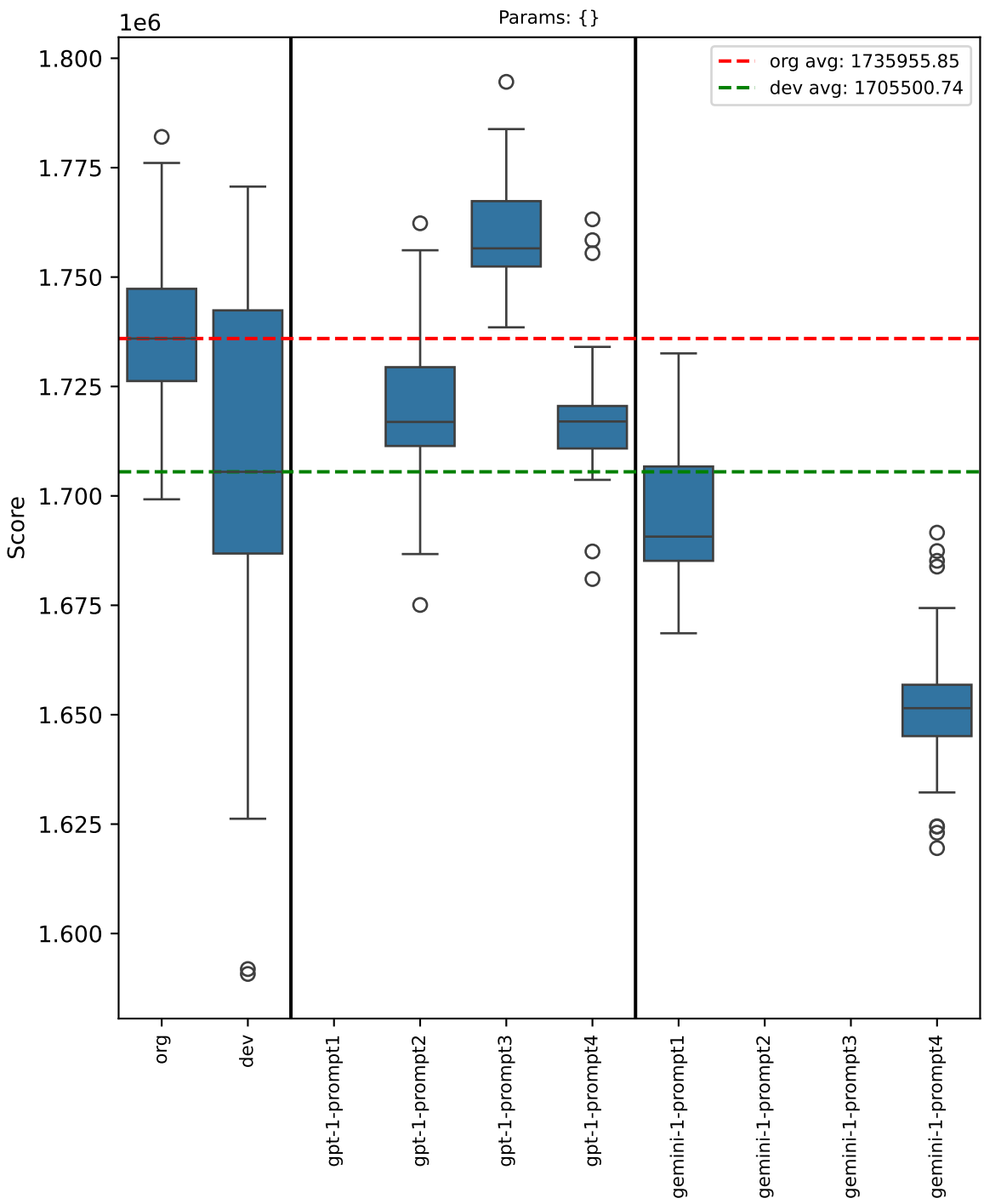




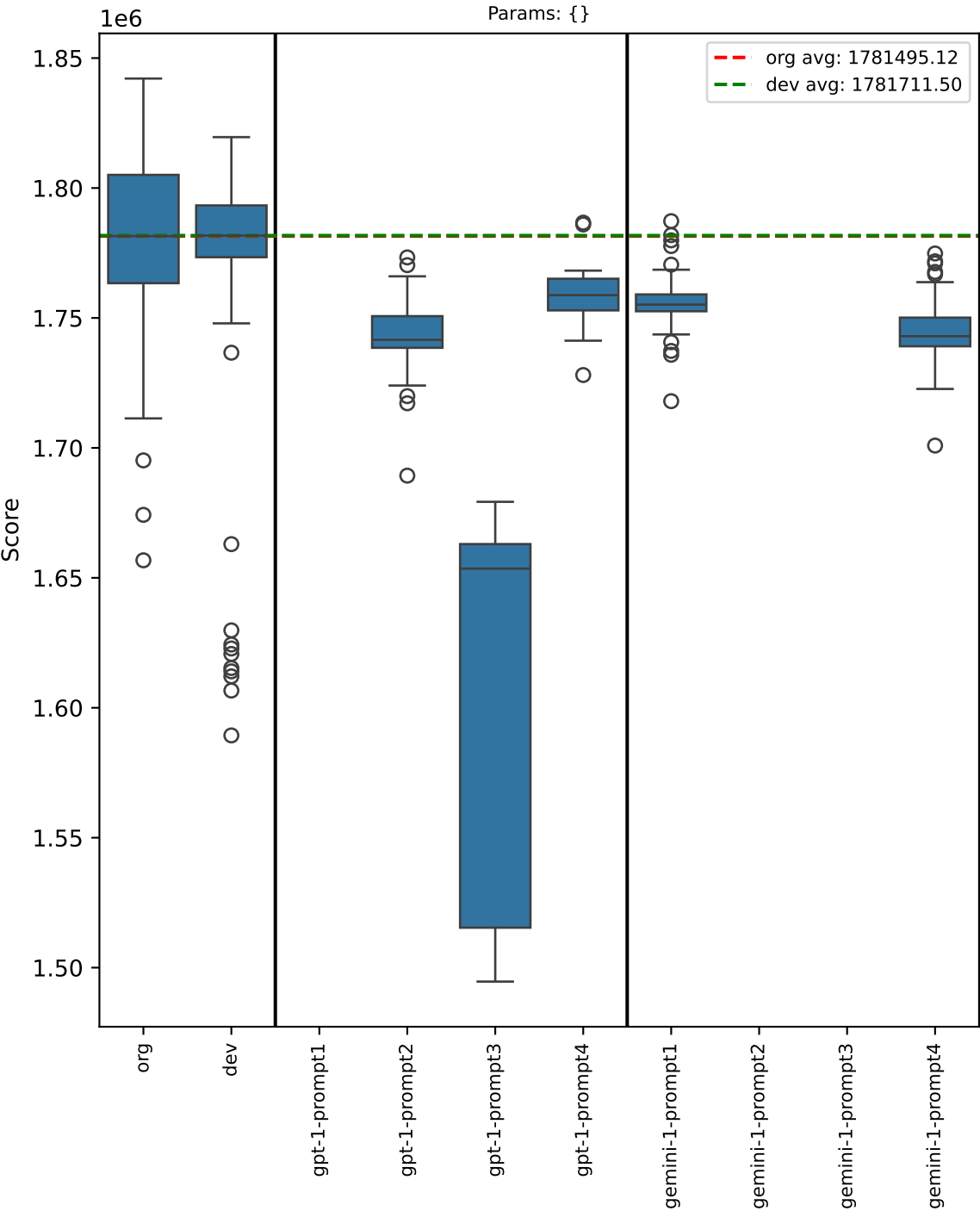
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeSimpleLineString



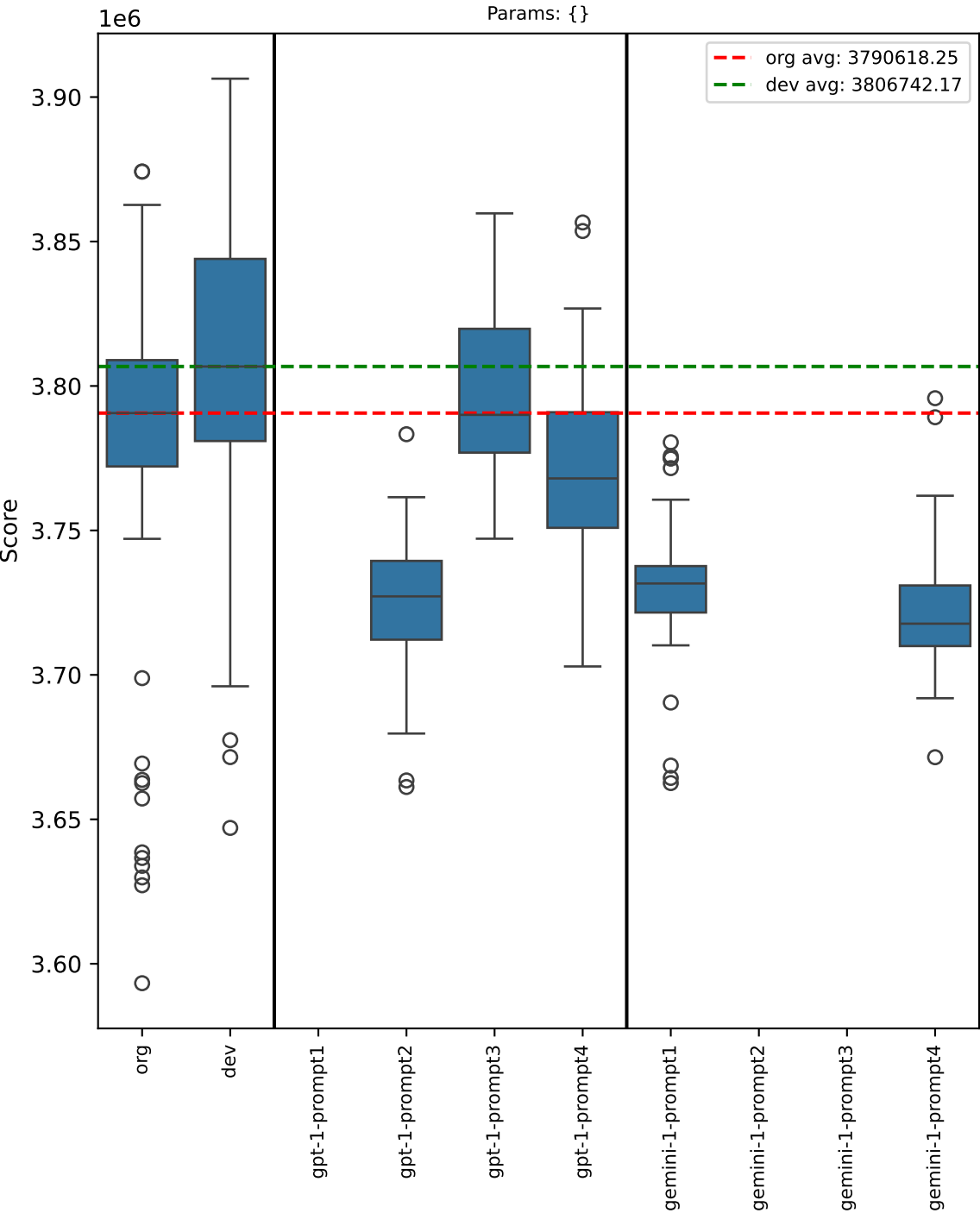
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeSimpleMultiLineString



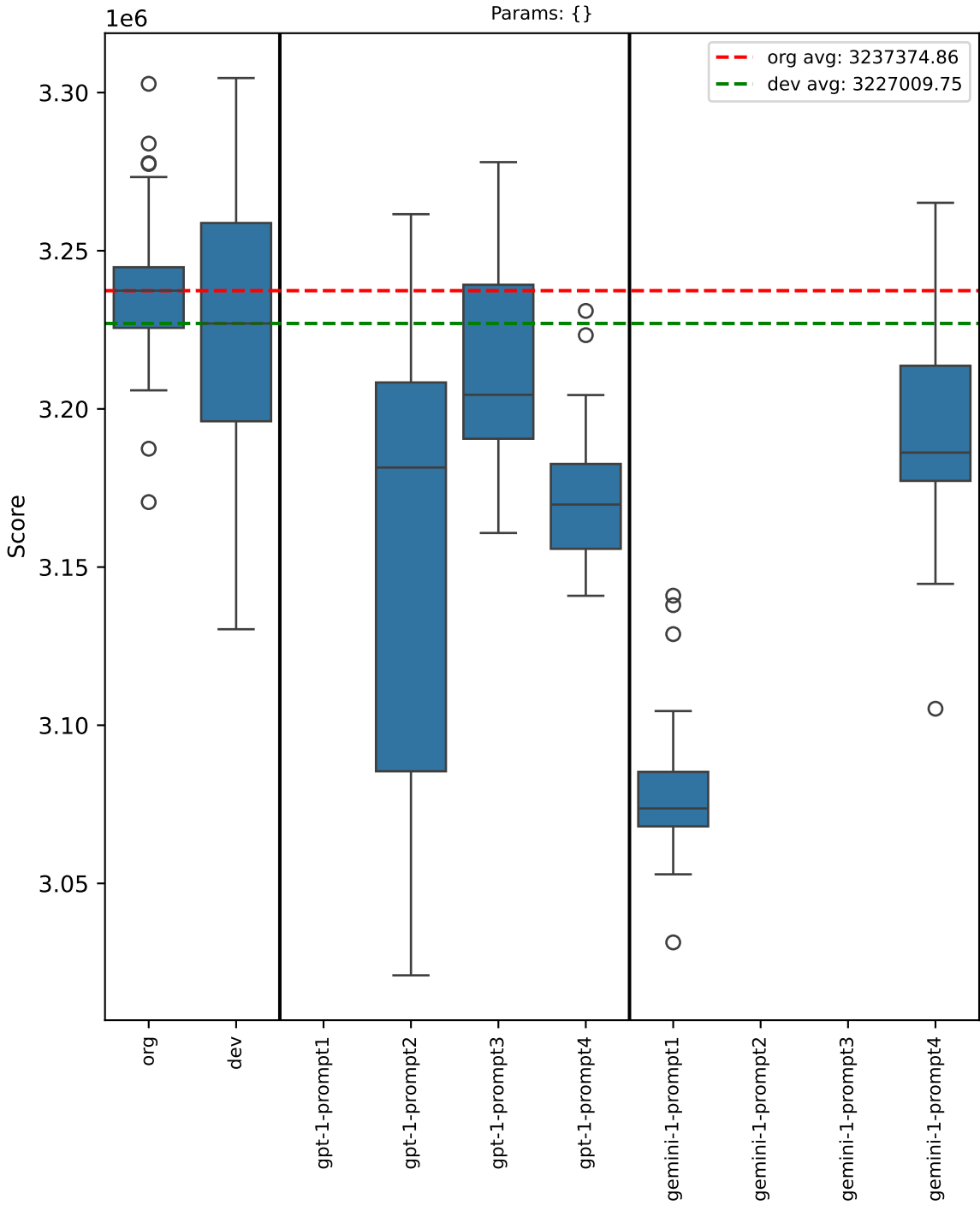
Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeSimpleMultiPolygon



Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeSimpleMultipoint

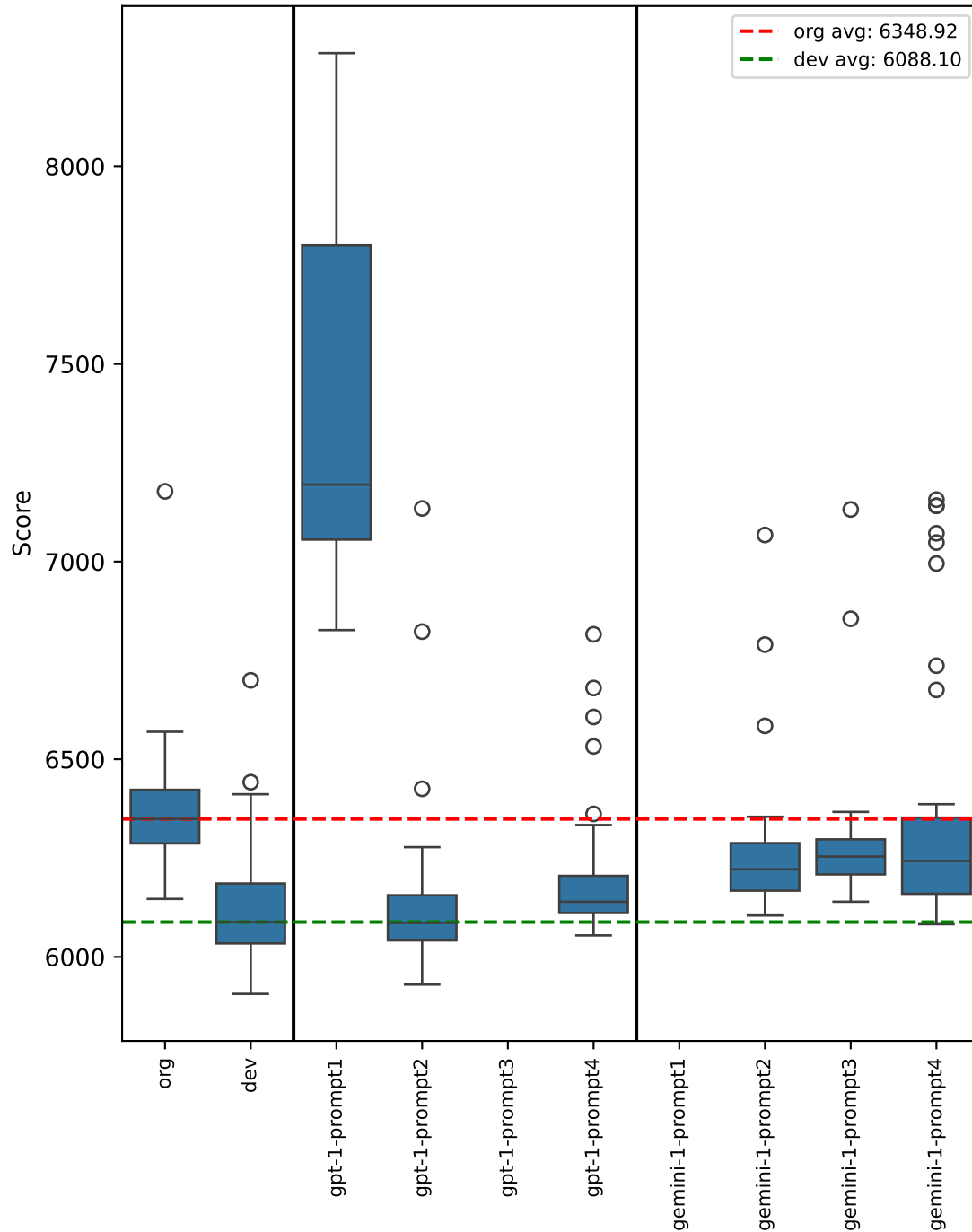


Benchmark: com.facebook.presto.geospatial.BenchmarkGeometrySerde.serializeSimplePolygon

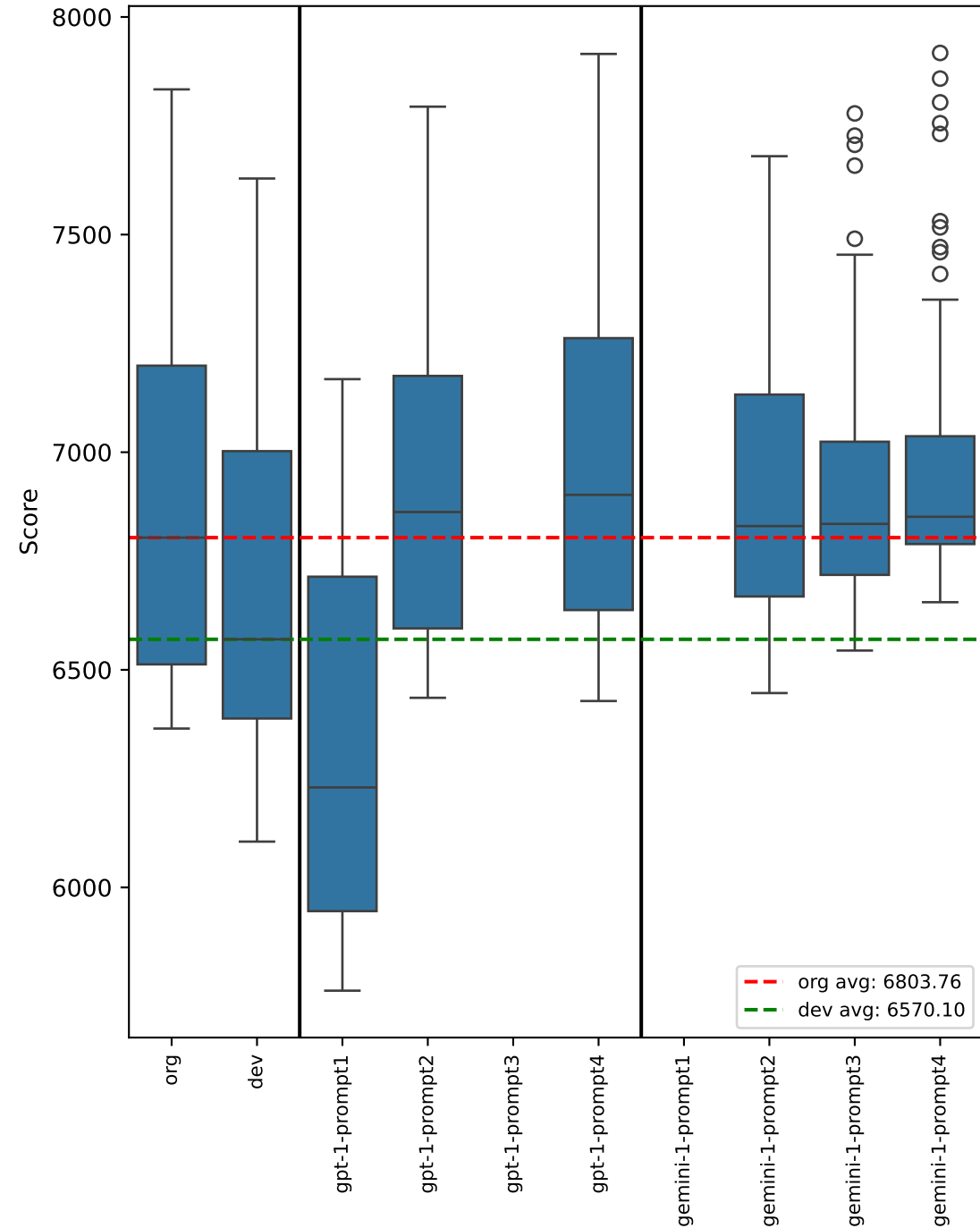


Benchmark: com.facebook.presto.operator.BenchmarkPartitionedOutputOperator.addPage

Params: {'enableCompression': 'true'}

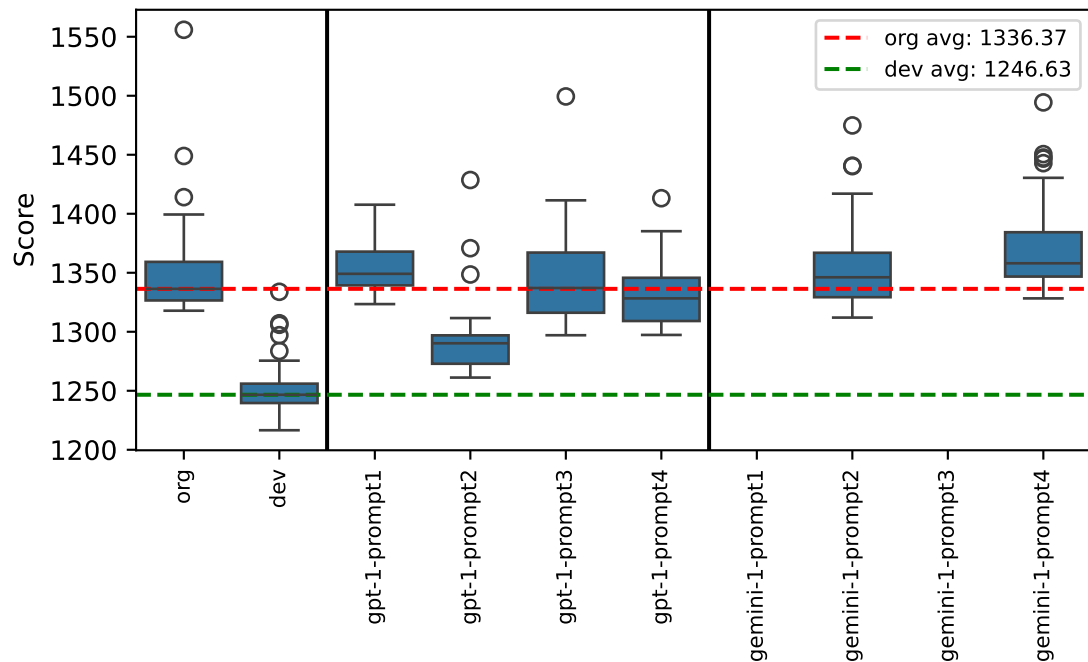


Params: {'enableCompression': 'false'}

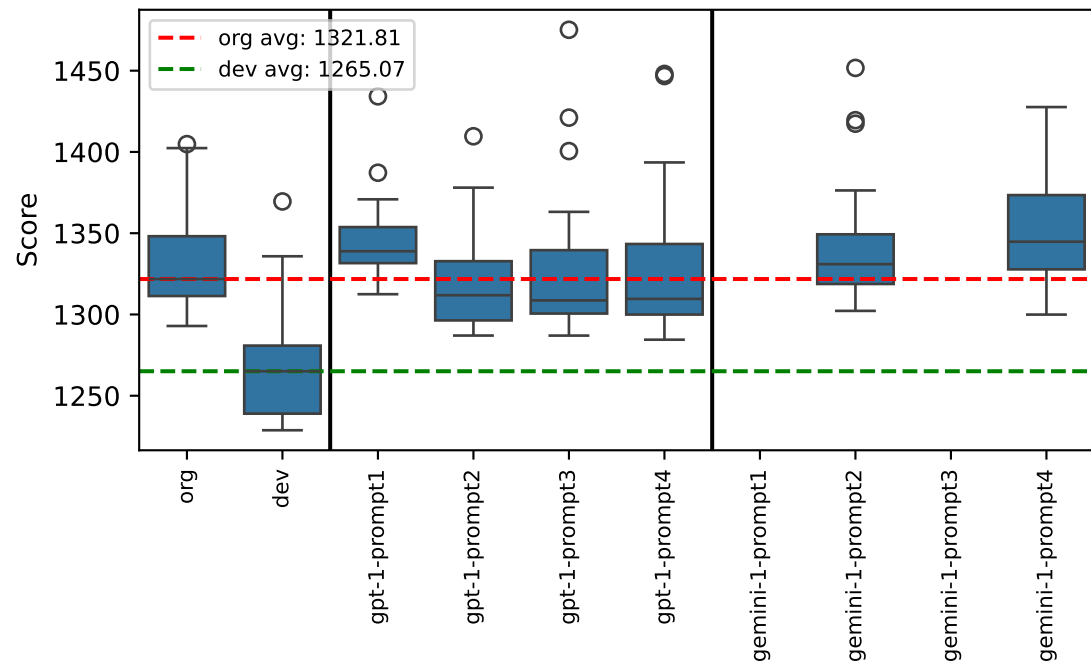


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

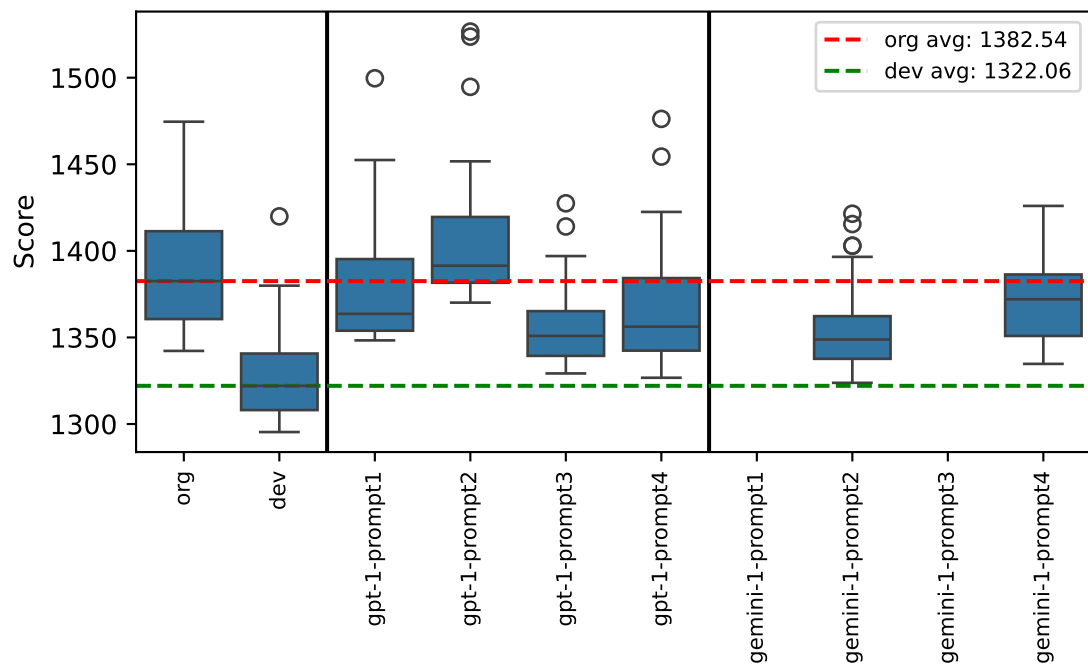
Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '10', 'rowsPerPartition': '10'}



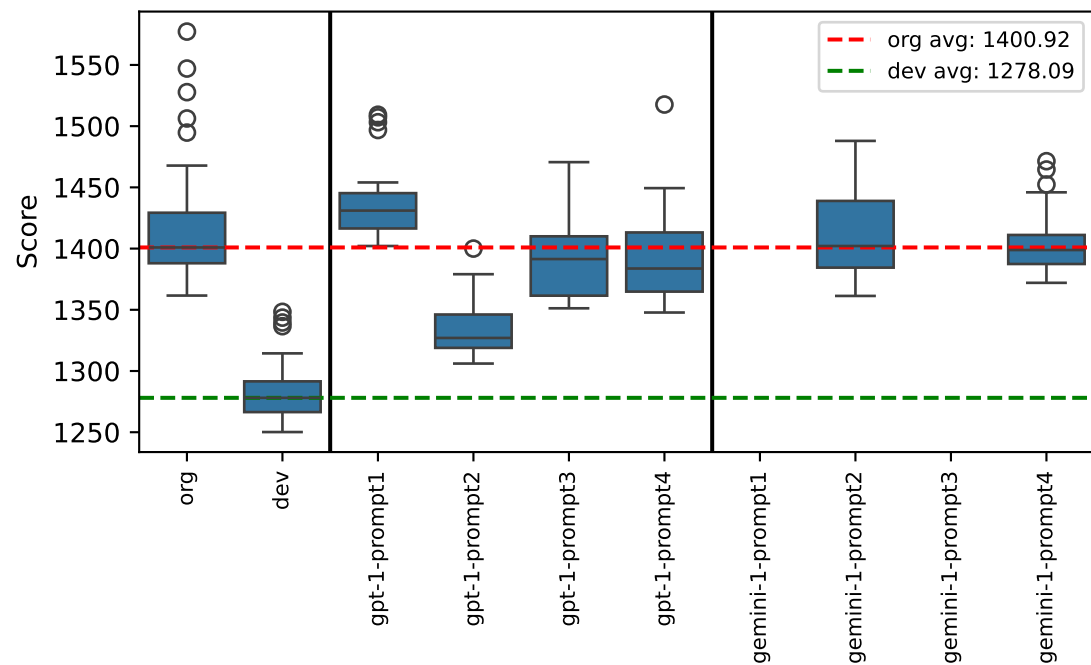
Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '10', 'rowsPerPartition': '20'}



Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '10', 'rowsPerPartition': '100'}

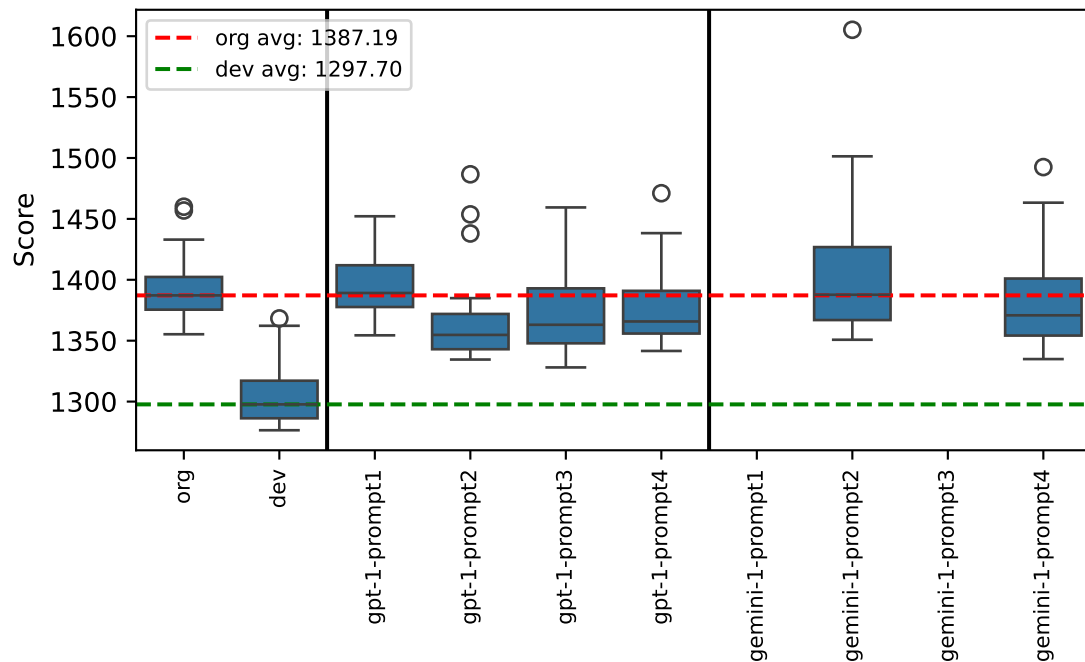


Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '50', 'rowsPerPartition': '10'}

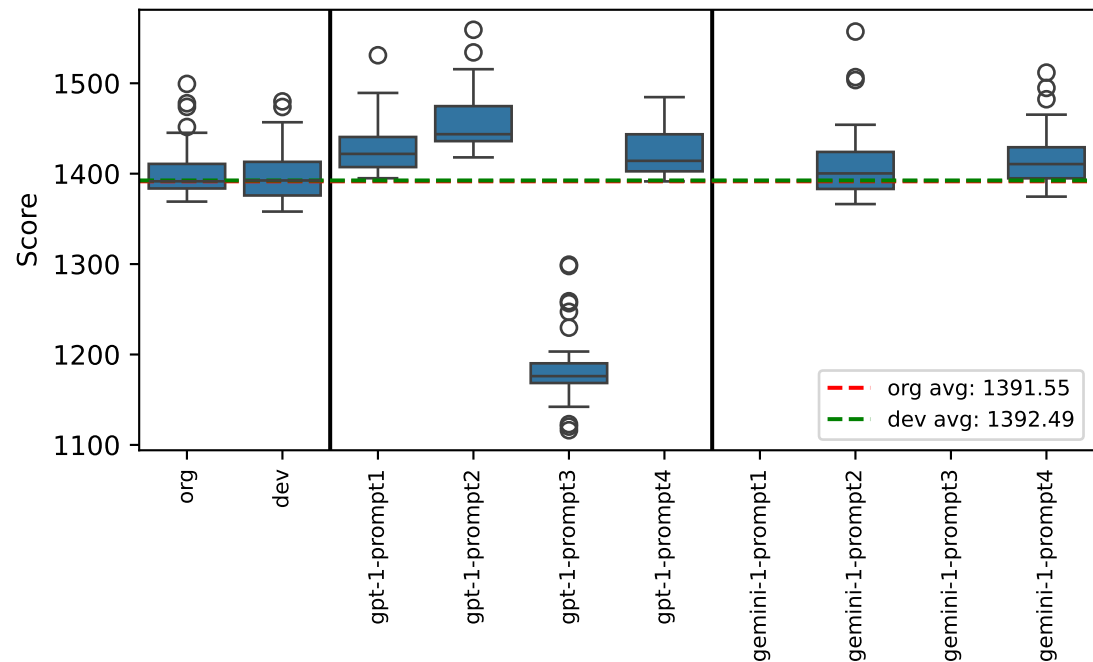


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

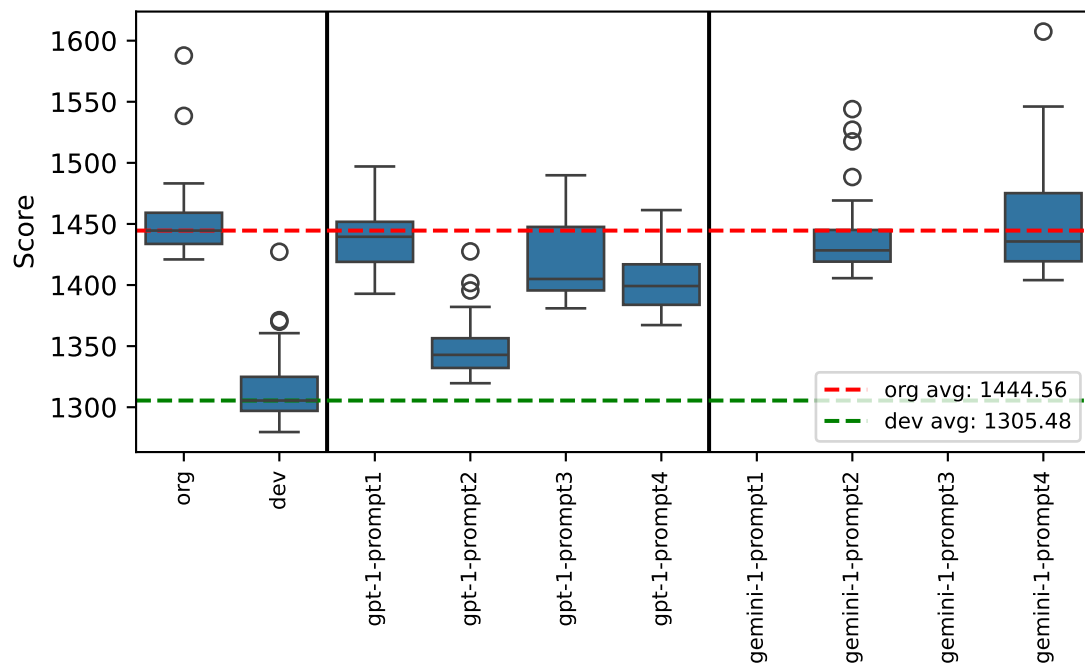
Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '50', 'rowsPerPartition': '20'}



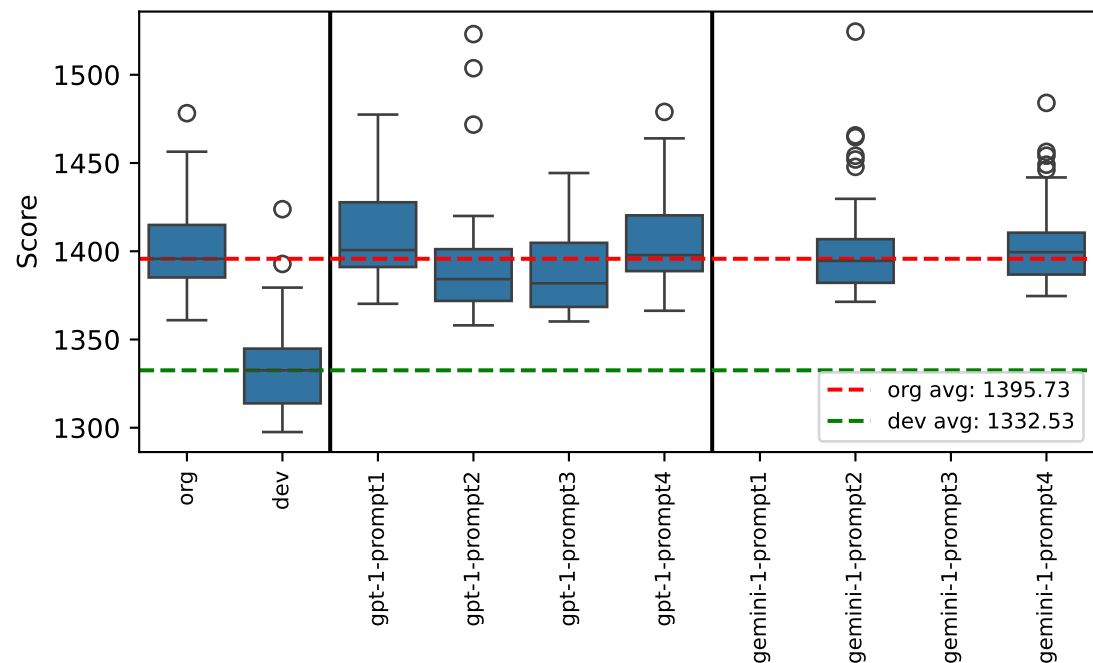
Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '50', 'rowsPerPartition': '100'}



Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '100', 'rowsPerPartition': '10'}

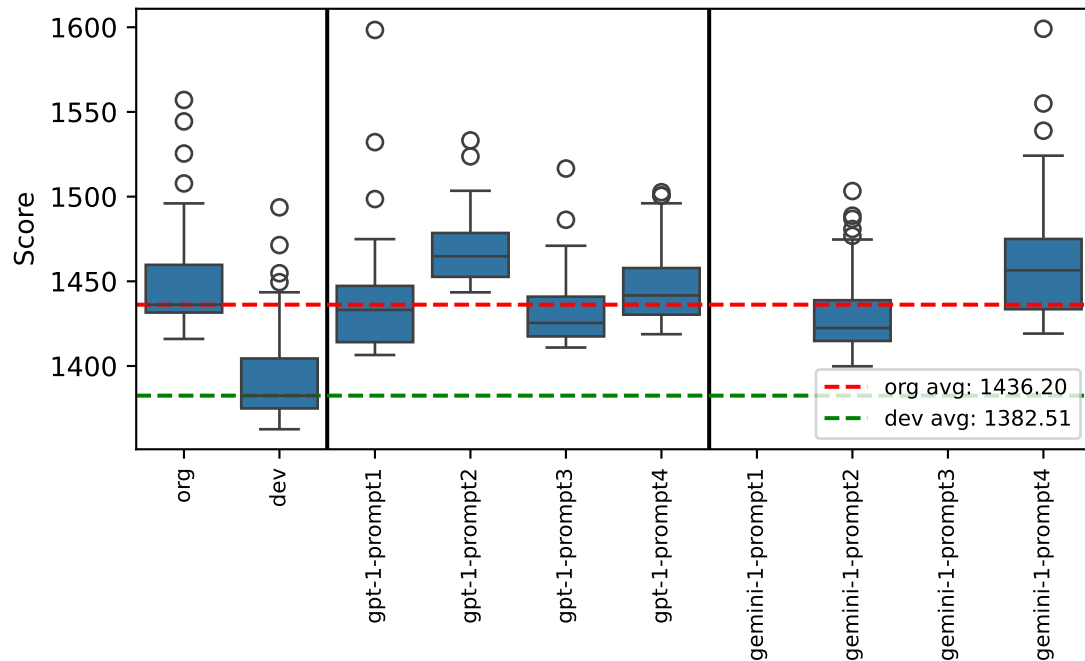


Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '100', 'rowsPerPartition': '20'}

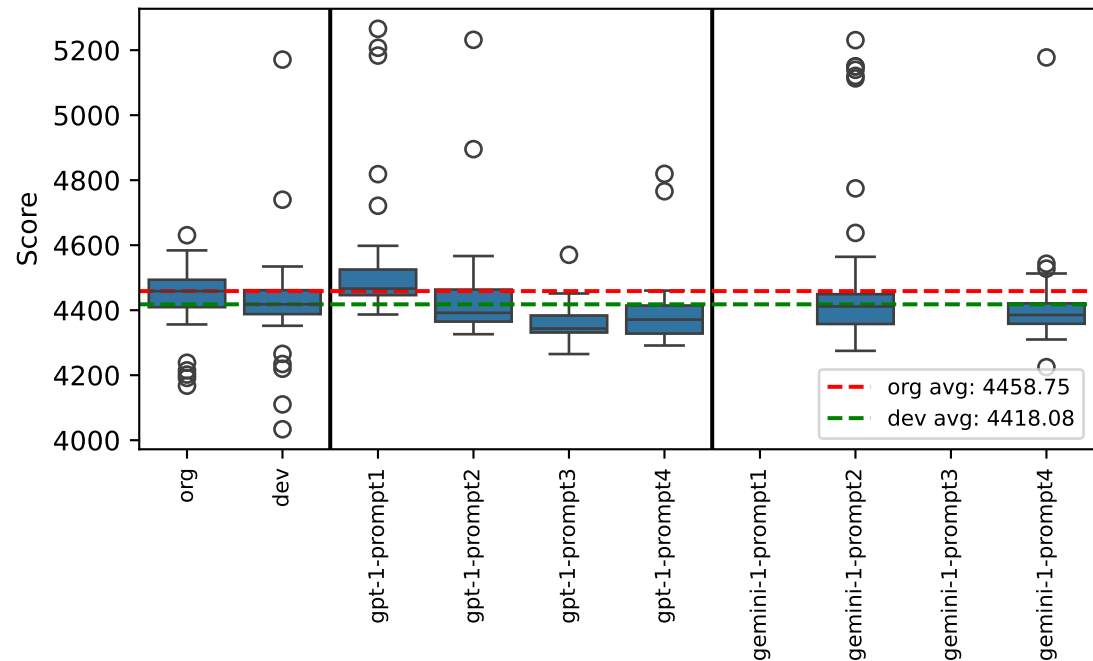


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

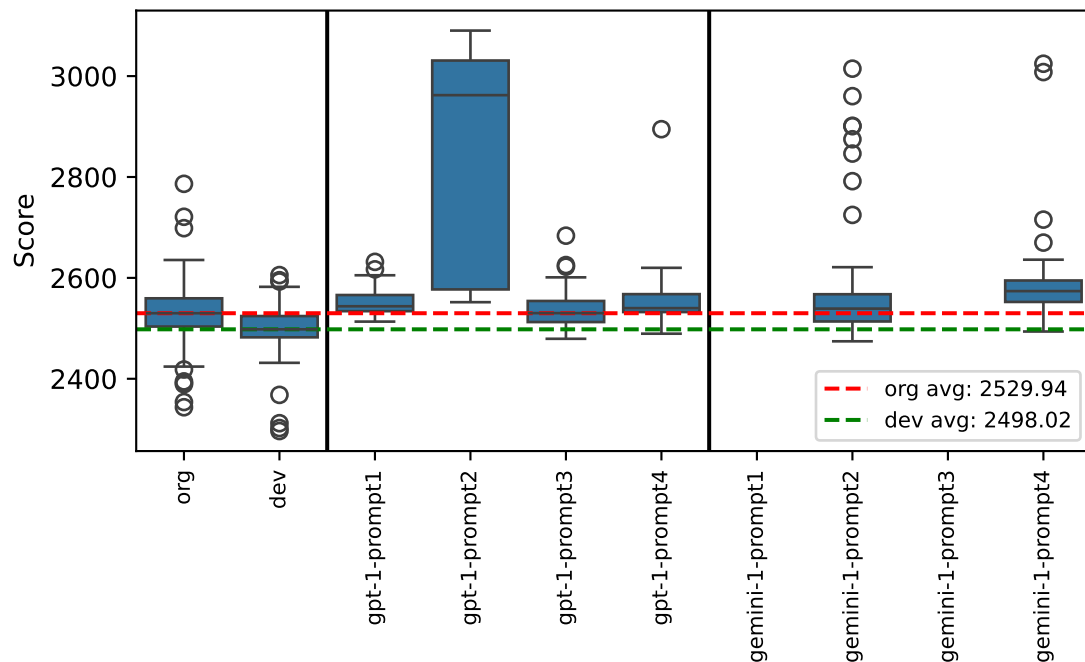
Params: {'numberOfPregroupedColumns': '0', 'partitionsPerGroup': '100', 'rowsPerPartition': '100'}



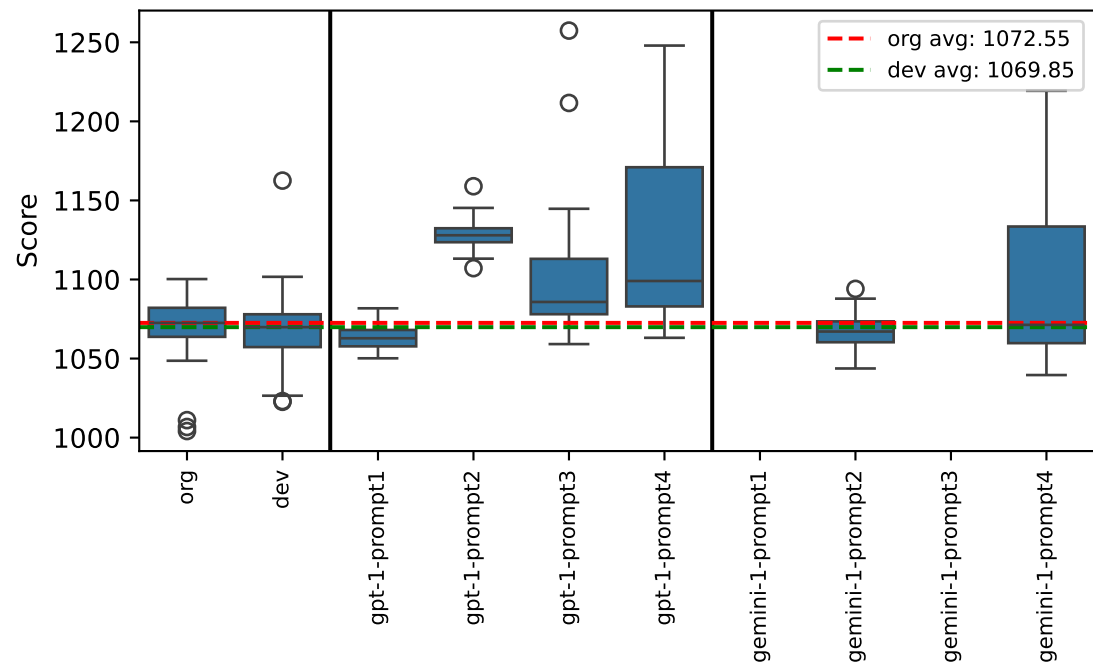
Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '10', 'rowsPerPartition': '10'}



Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '10', 'rowsPerPartition': '20'}

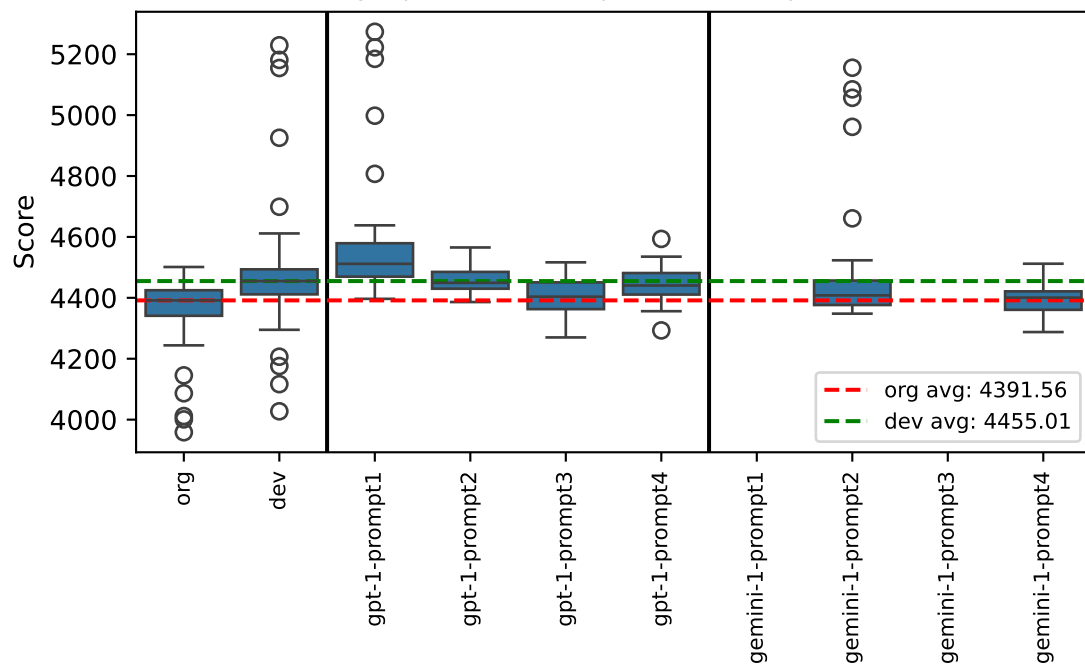


Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '10', 'rowsPerPartition': '100'}

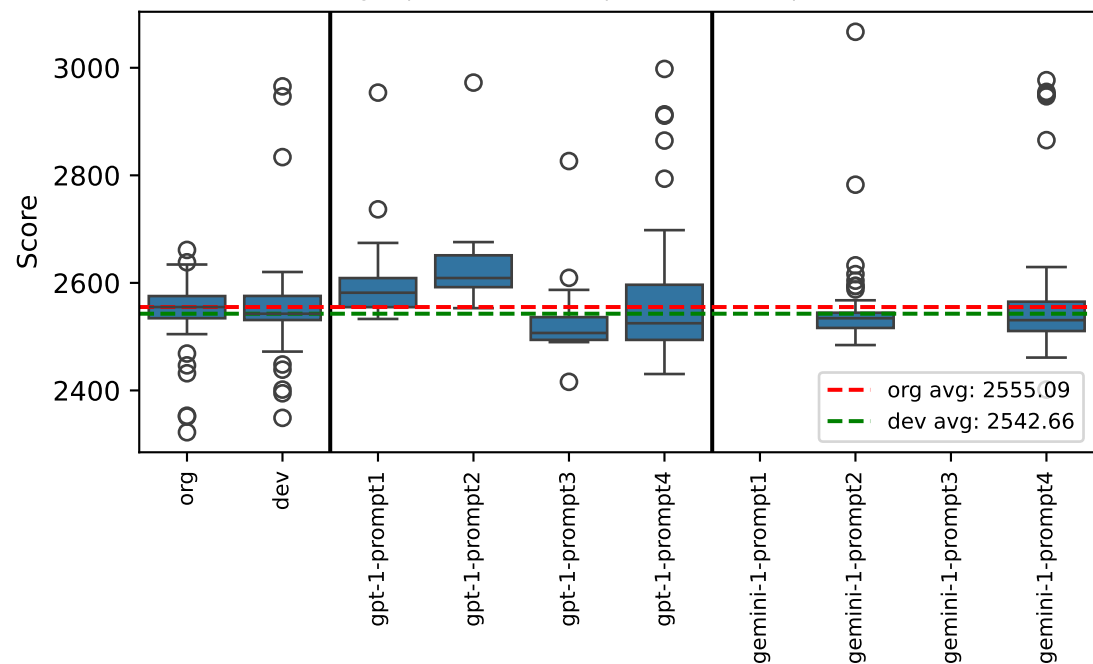


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

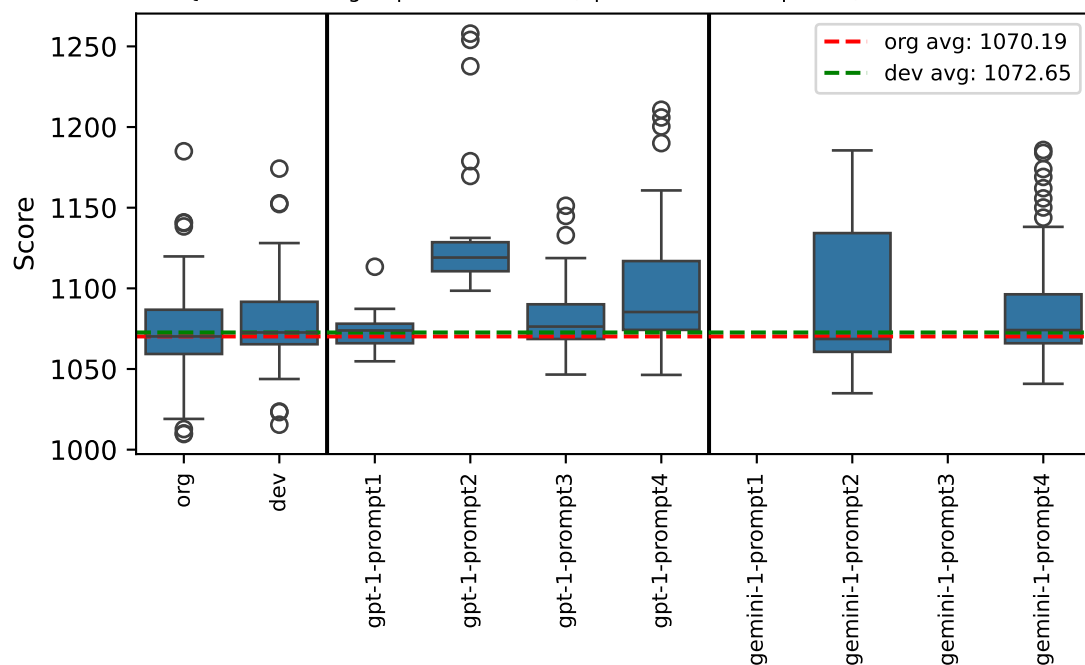
Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '50', 'rowsPerPartition': '10'}



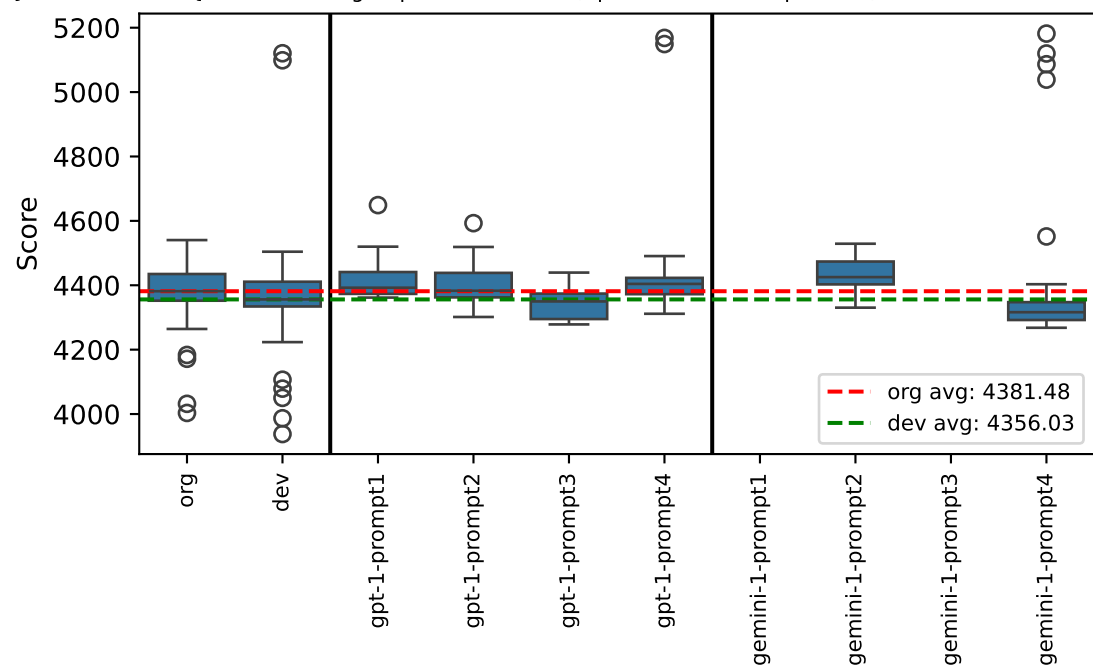
Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '50', 'rowsPerPartition': '20'}



Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '50', 'rowsPerPartition': '100'}

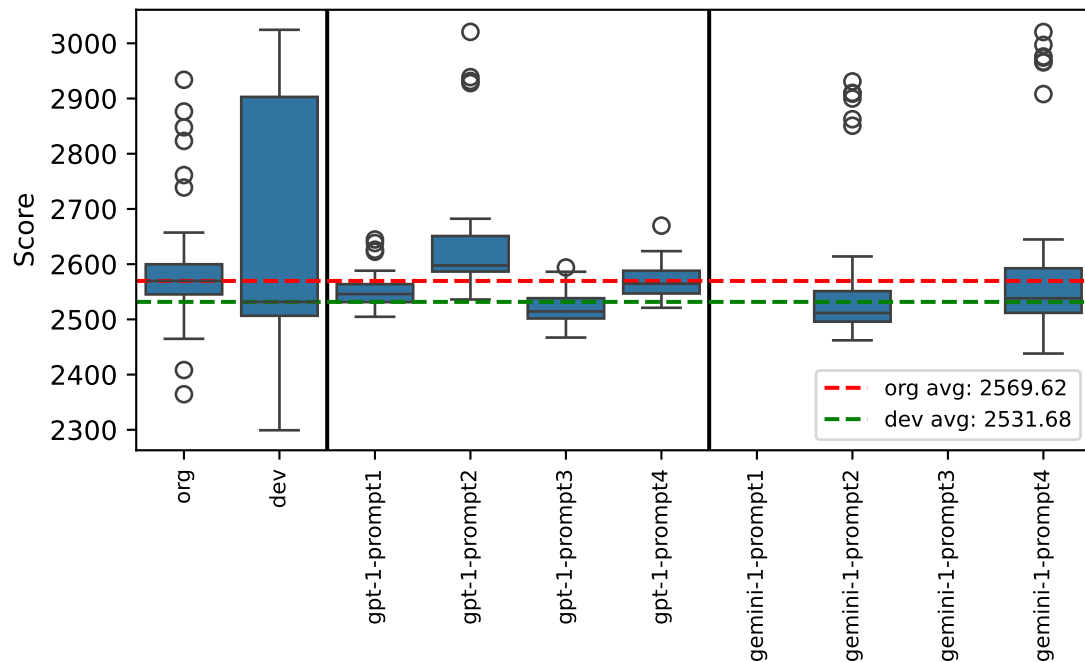


Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '100', 'rowsPerPartition': '10'}

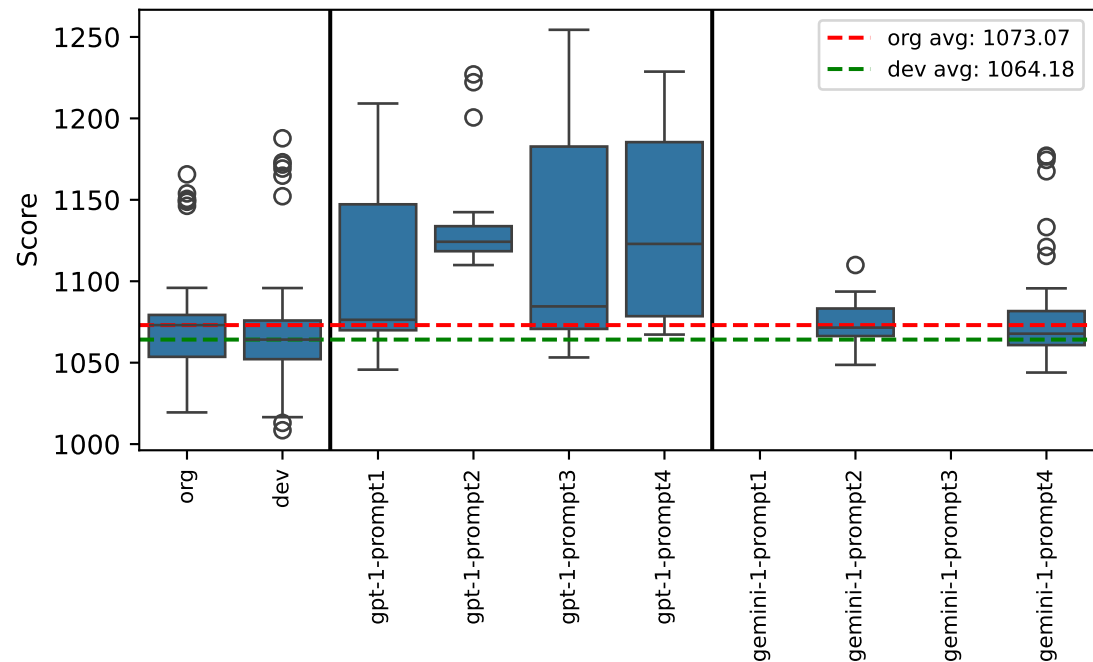


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

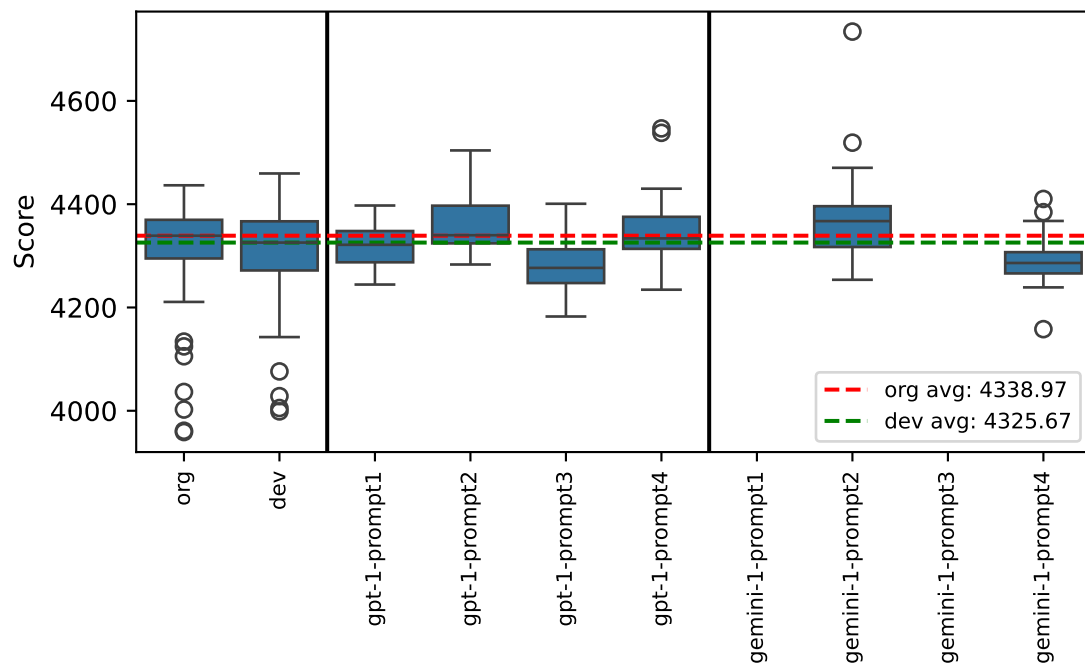
Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '100', 'rowsPerPartition': '20'}



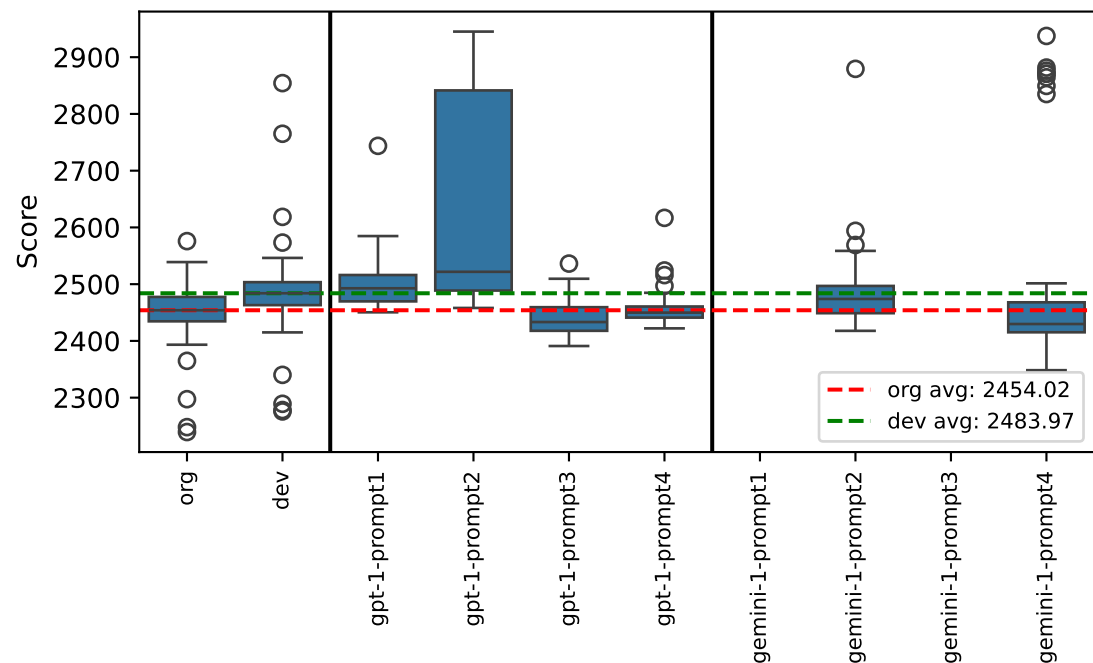
Params: {'numberOfPregroupedColumns': '1', 'partitionsPerGroup': '100', 'rowsPerPartition': '100'}



Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '10', 'rowsPerPartition': '10'}

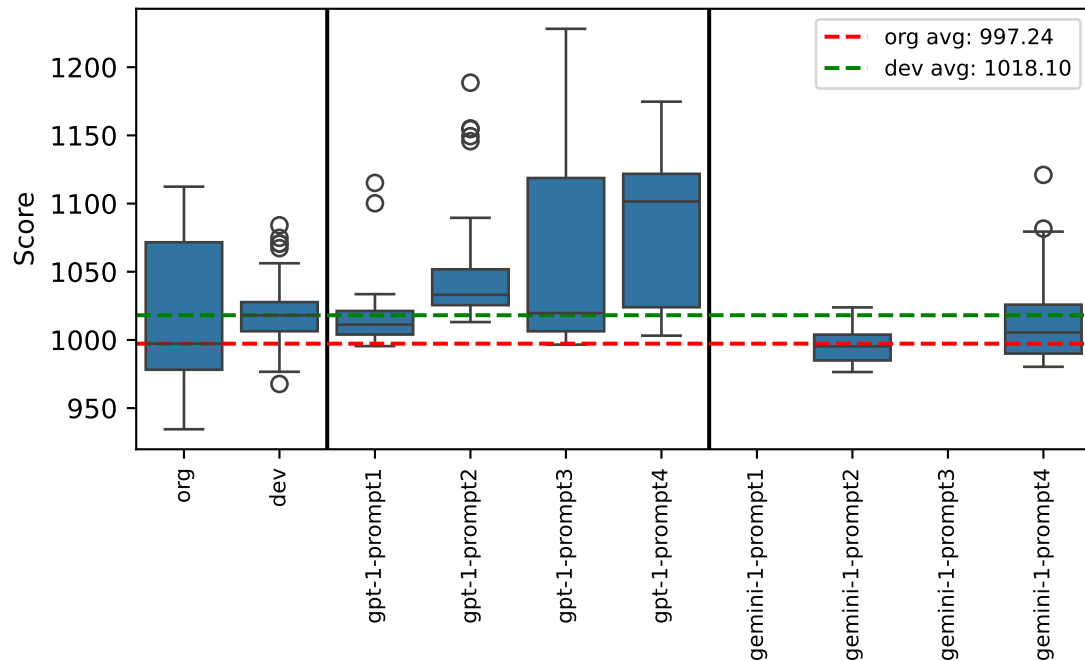


Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '10', 'rowsPerPartition': '20'}

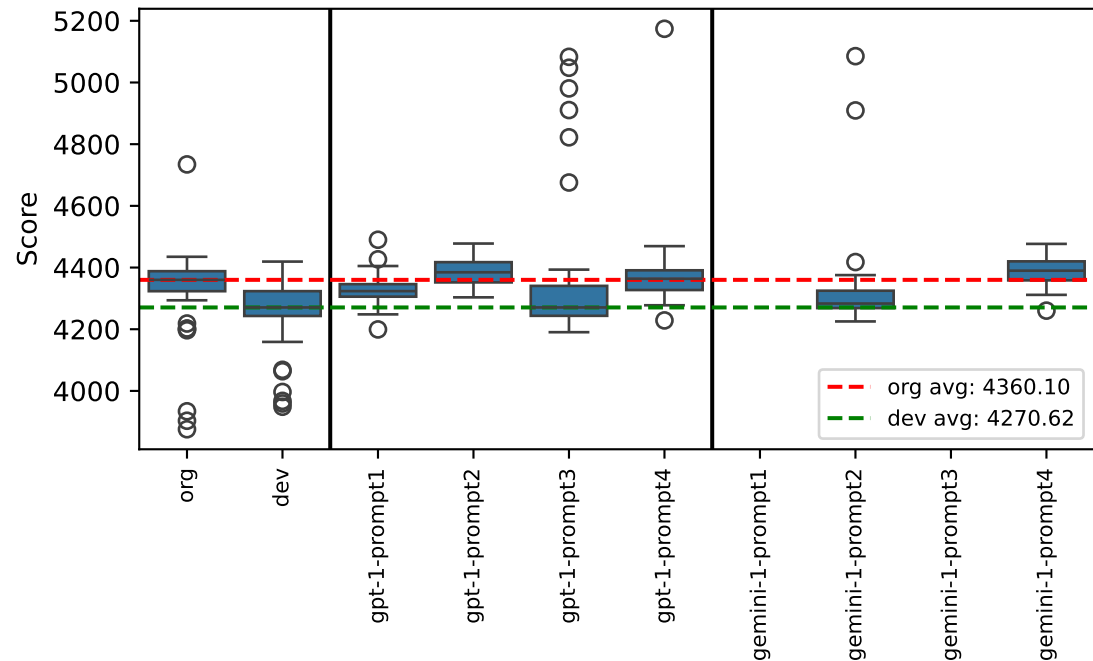


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

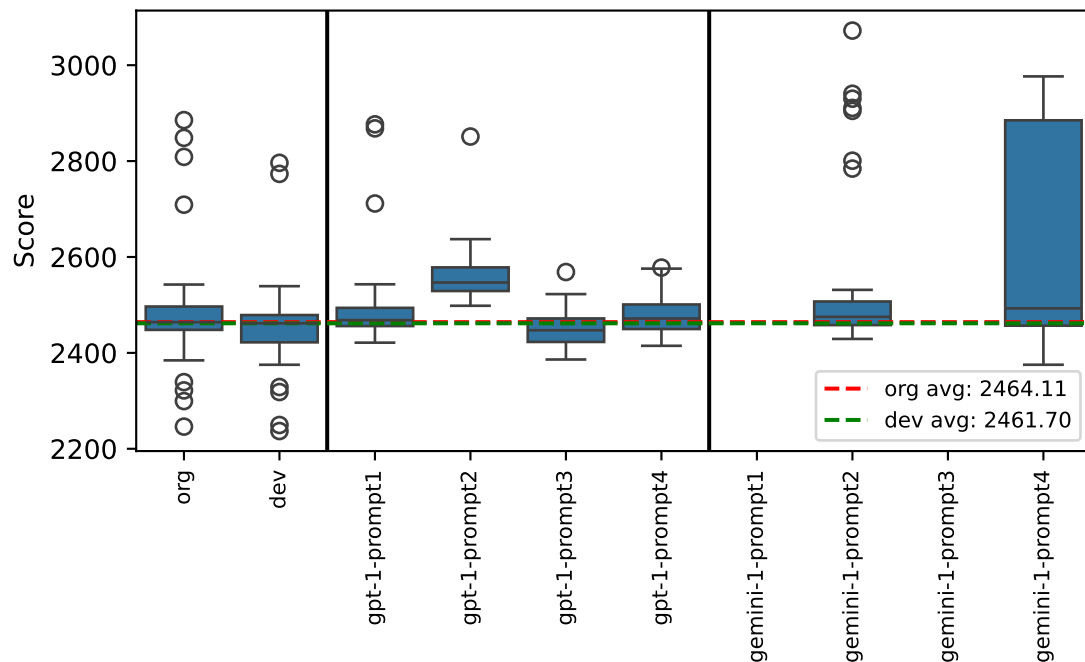
Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '10', 'rowsPerPartition': '100'}



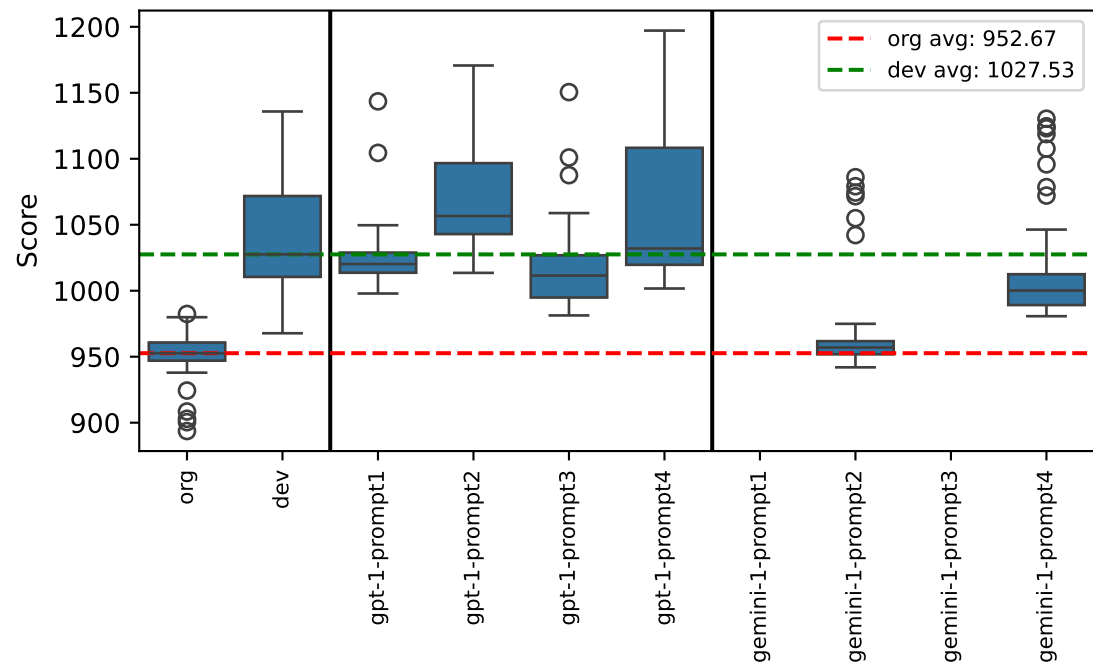
Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '50', 'rowsPerPartition': '10'}



Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '50', 'rowsPerPartition': '20'}

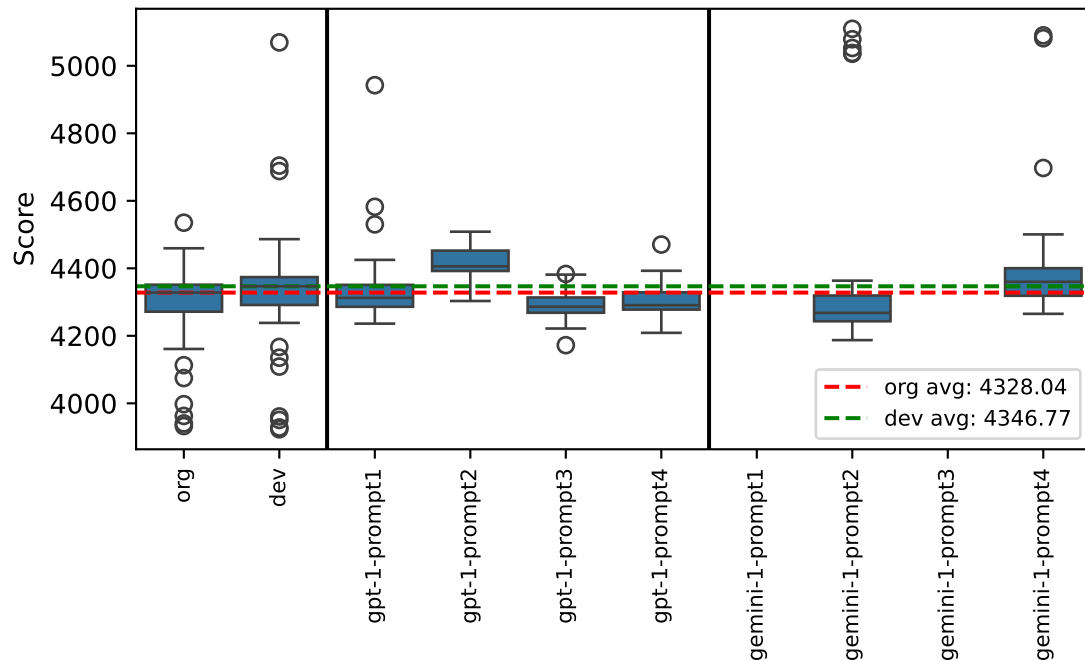


Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '50', 'rowsPerPartition': '100'}

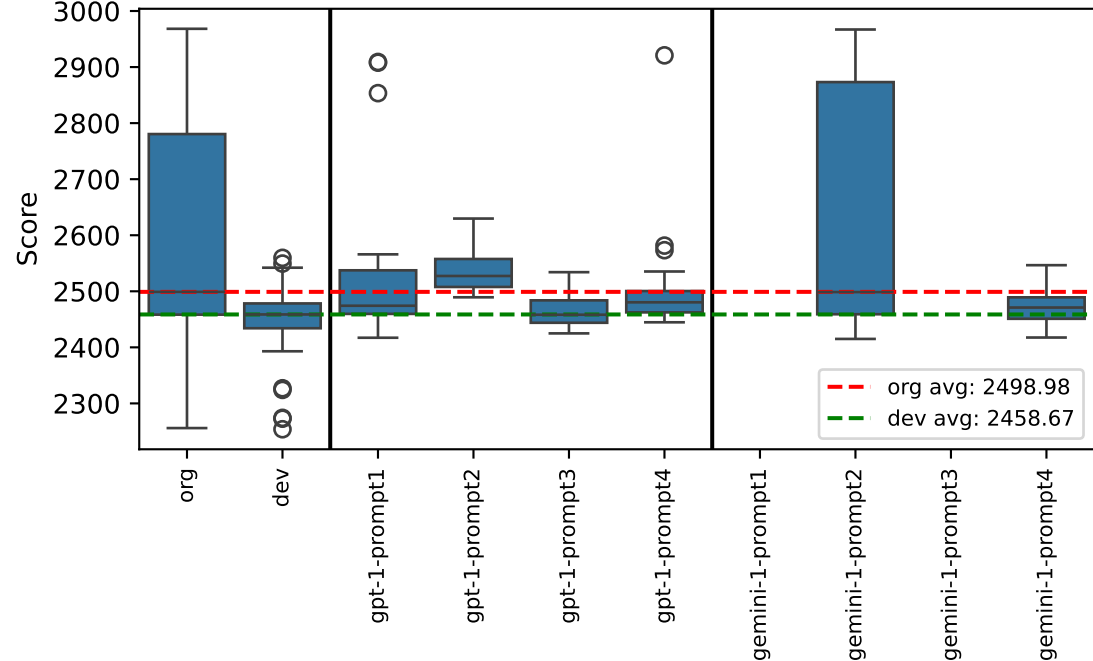


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

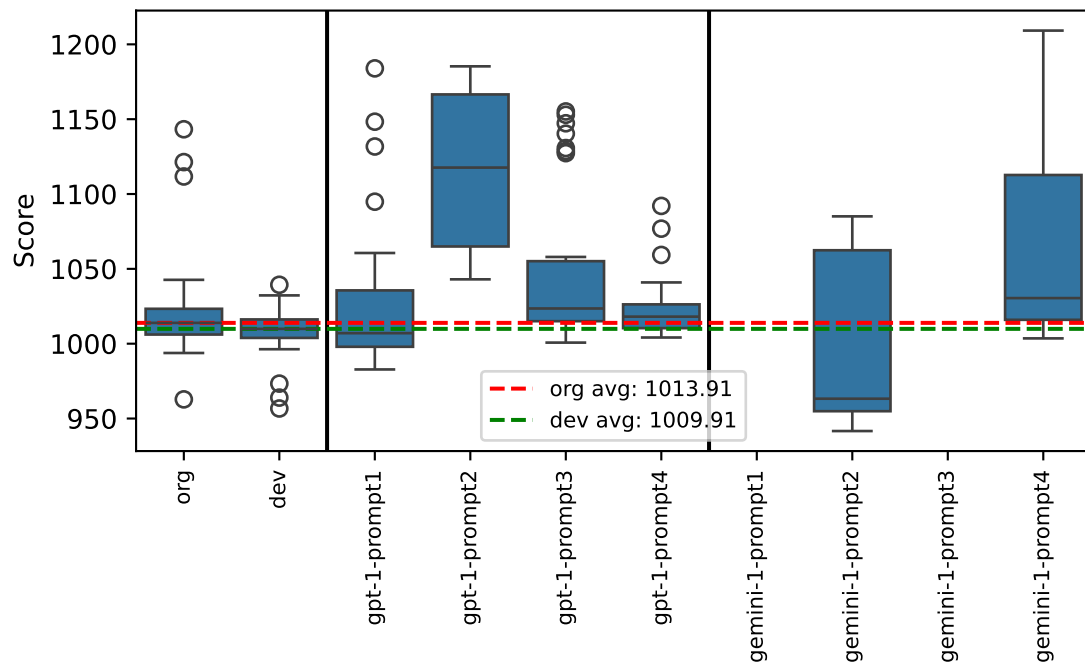
Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '100', 'rowsPerPartition': '10'}



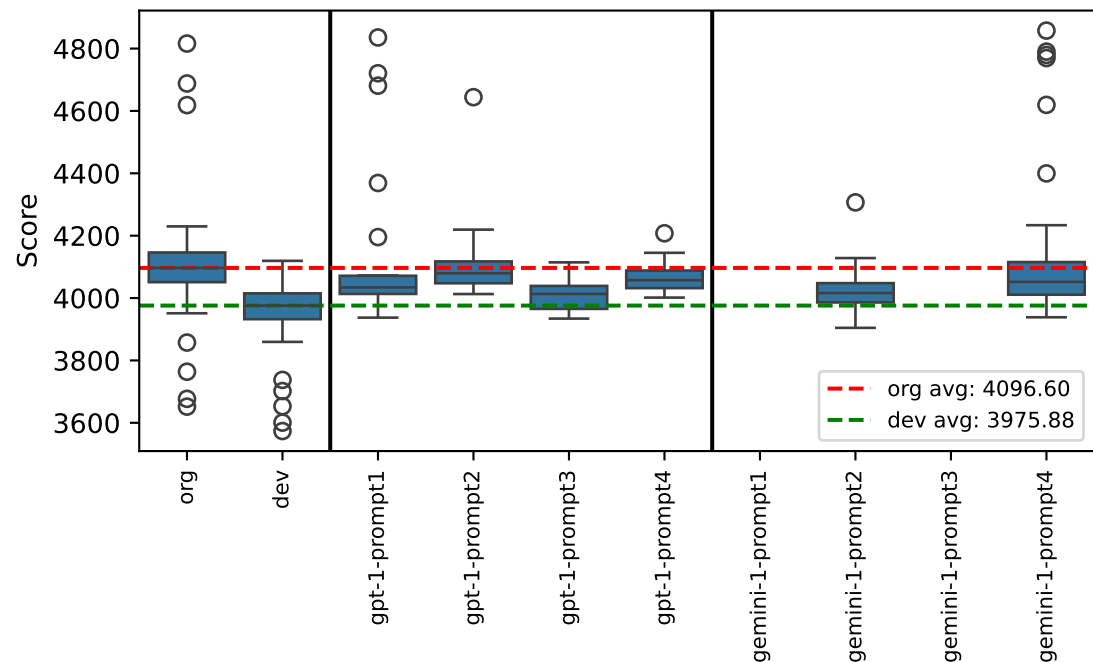
Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '100', 'rowsPerPartition': '20'}



Params: {'numberOfPregroupedColumns': '2', 'partitionsPerGroup': '100', 'rowsPerPartition': '100'}

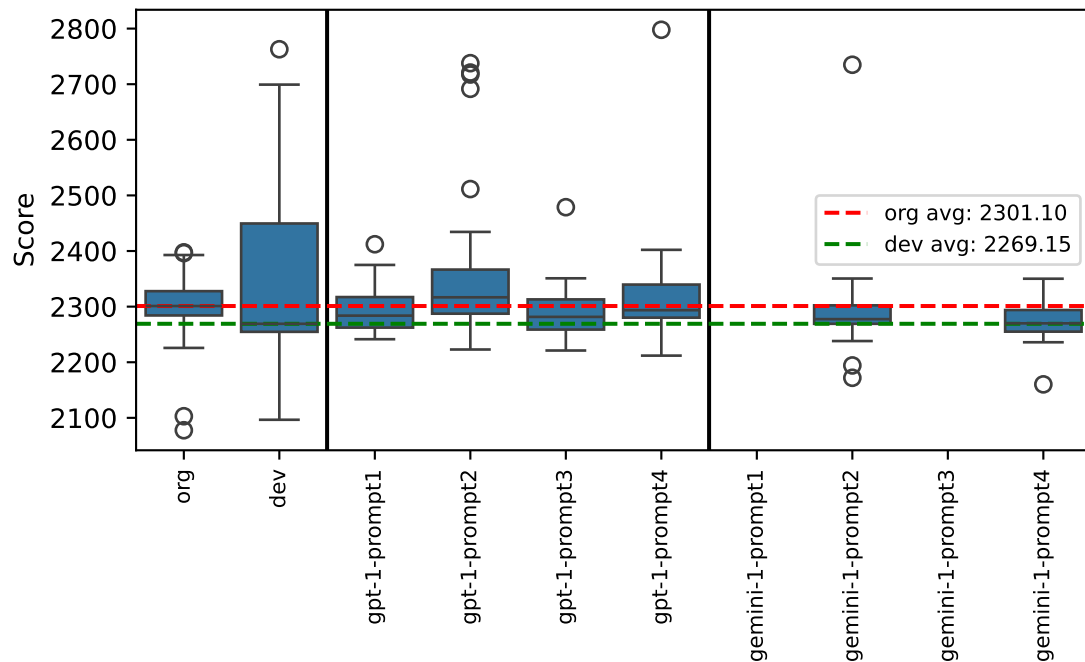


Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '10', 'rowsPerPartition': '10'}

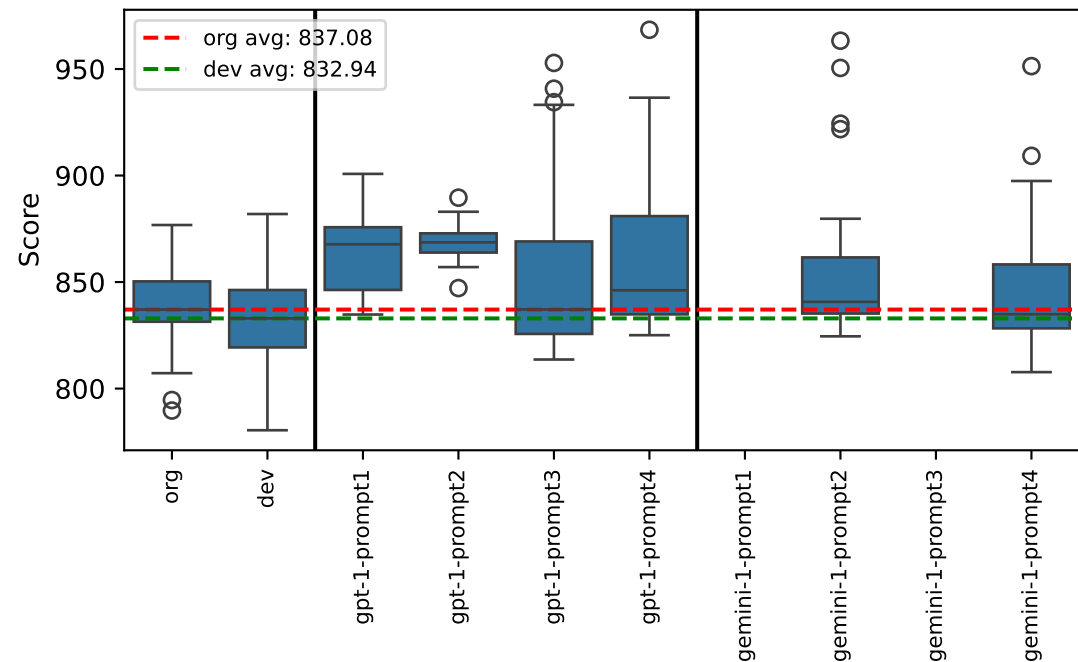


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

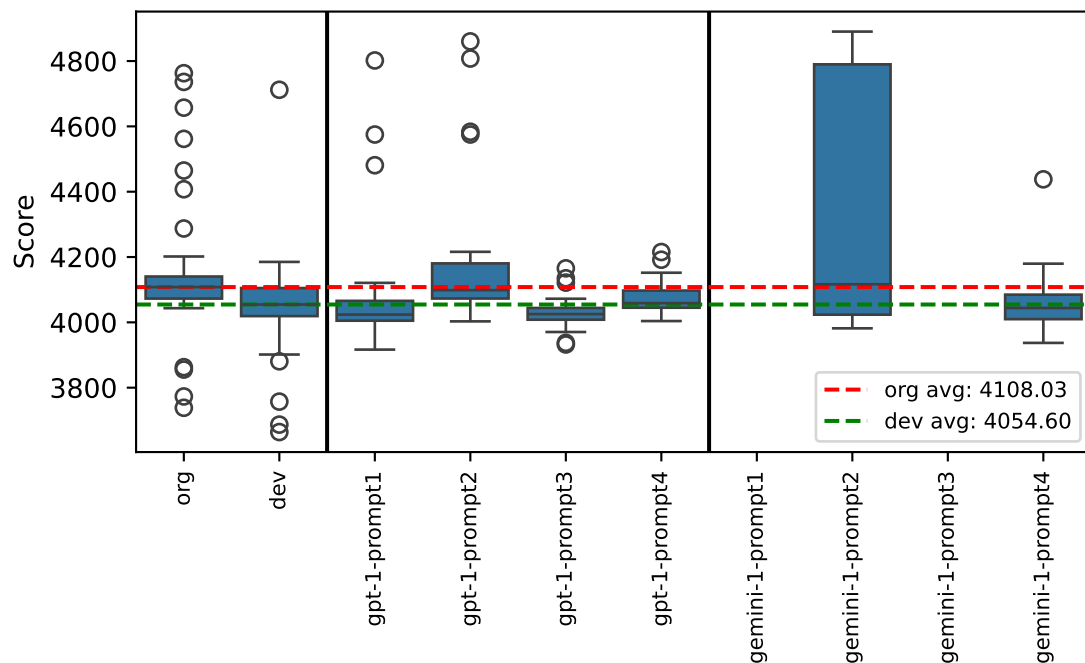
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '10', 'rowsPerPartition': '20'}



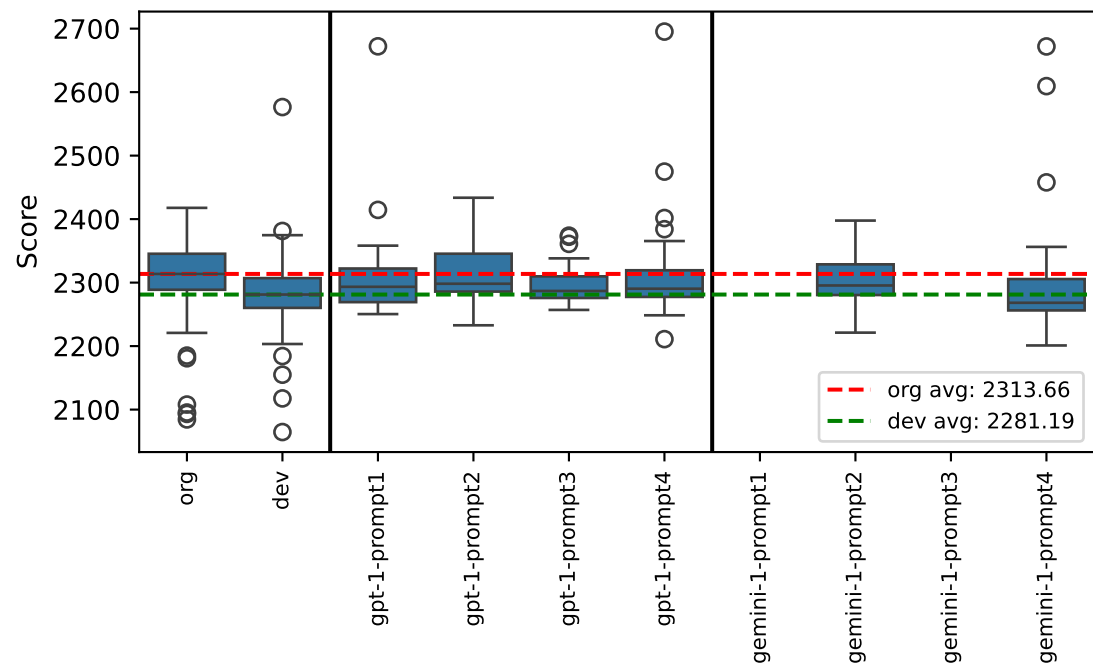
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '10', 'rowsPerPartition': '100'}



Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '50', 'rowsPerPartition': '10'}

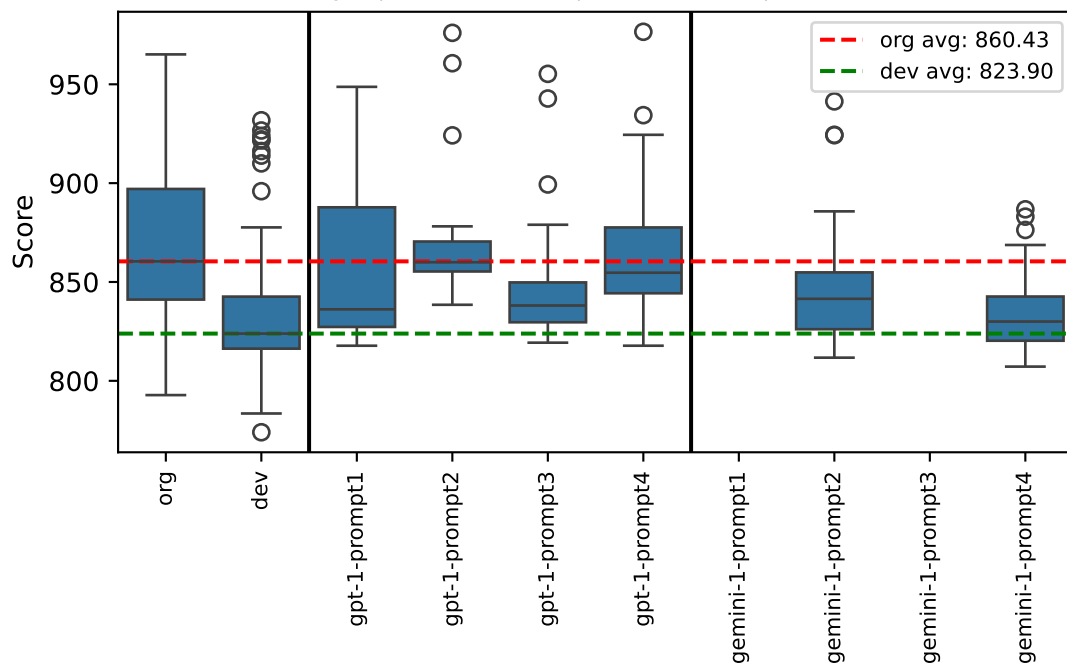


Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '50', 'rowsPerPartition': '20'}

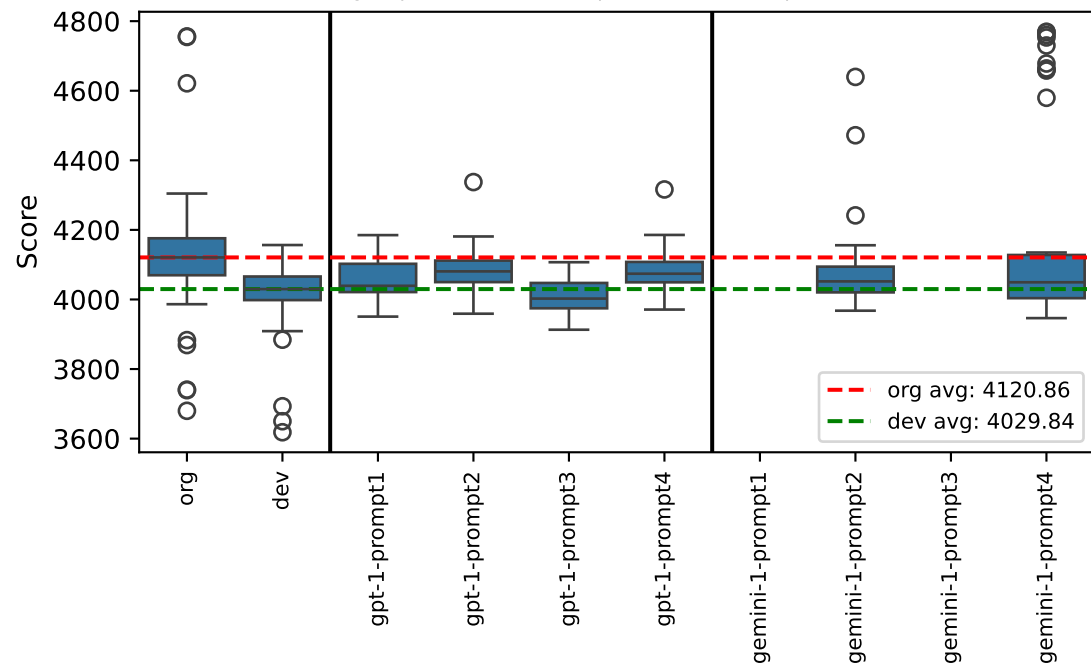


Benchmark: com.facebook.presto.operator.BenchmarkWindowOperator.benchmark

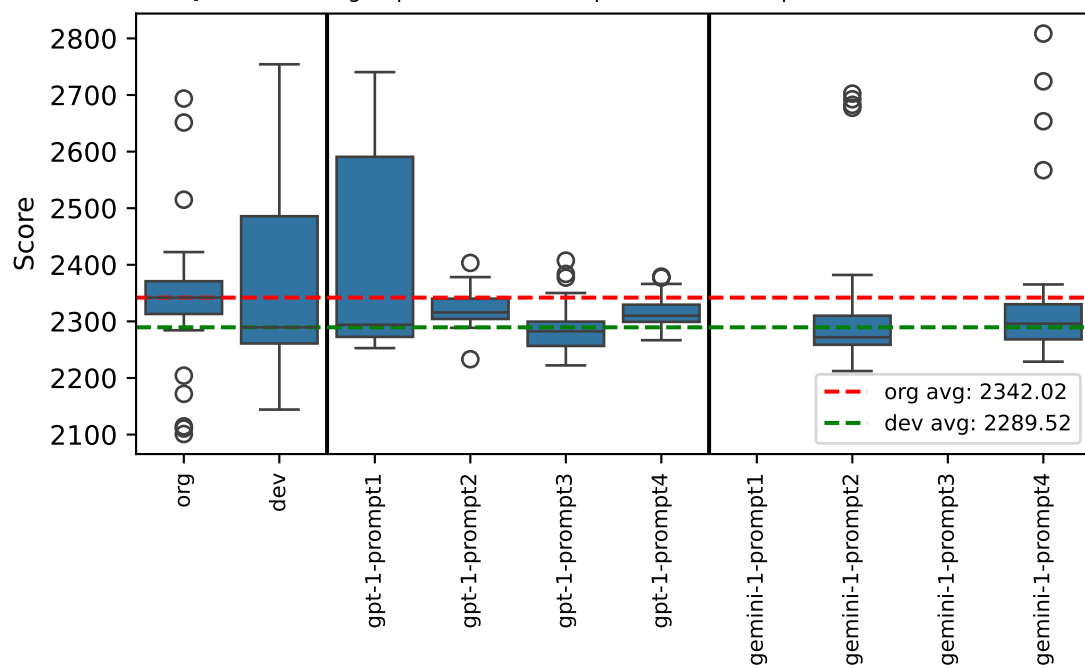
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '50', 'rowsPerPartition': '100'}



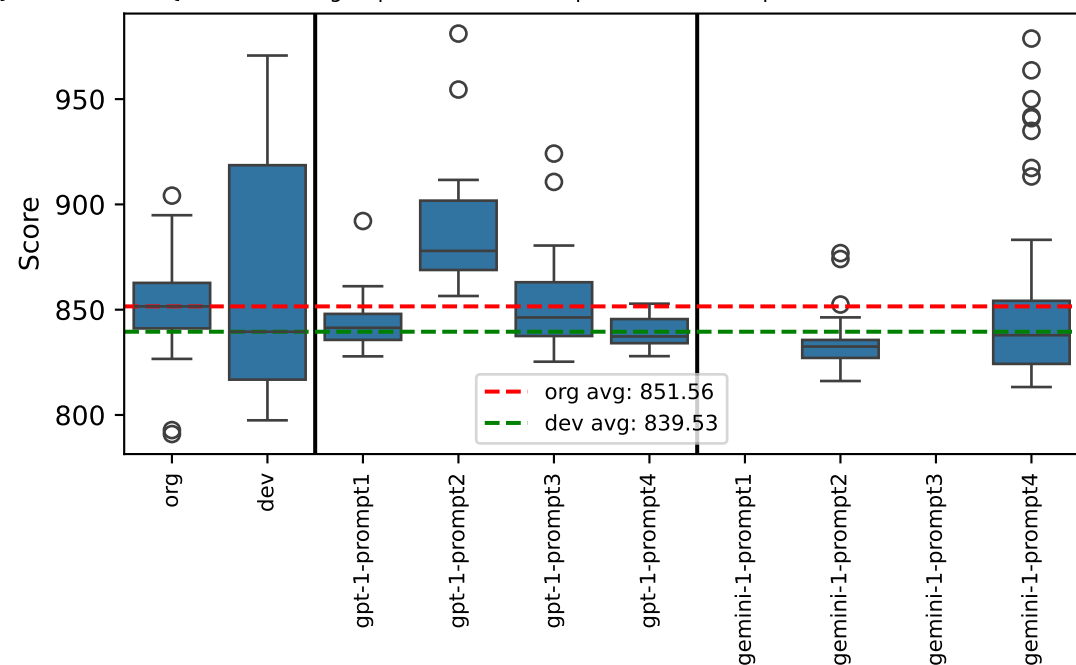
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '100', 'rowsPerPartition': '10'}



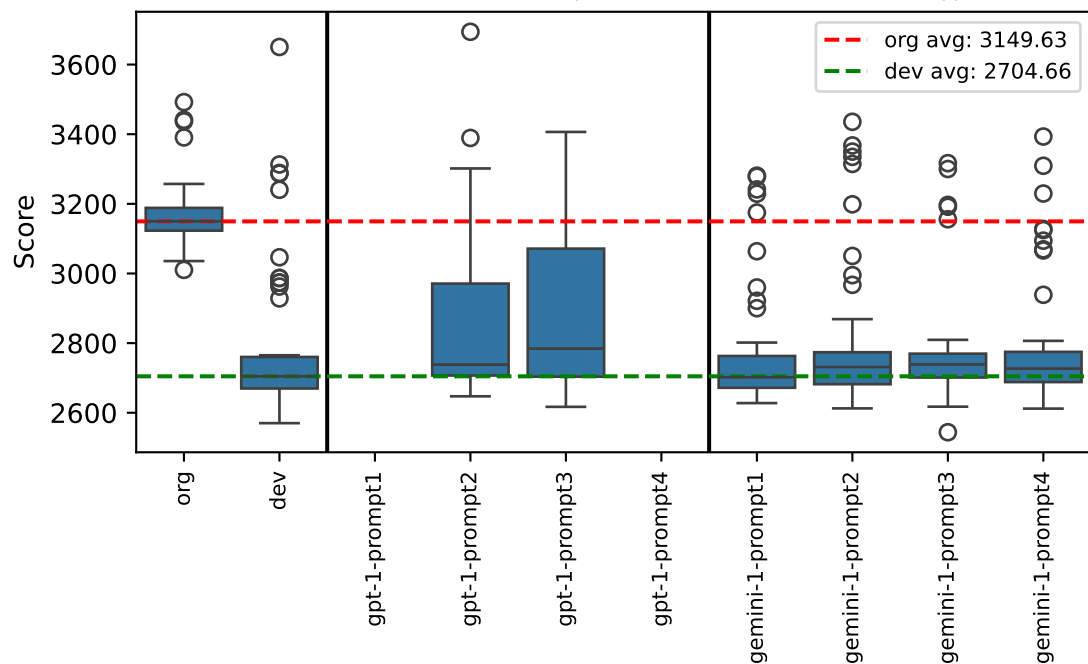
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '100', 'rowsPerPartition': '20'}



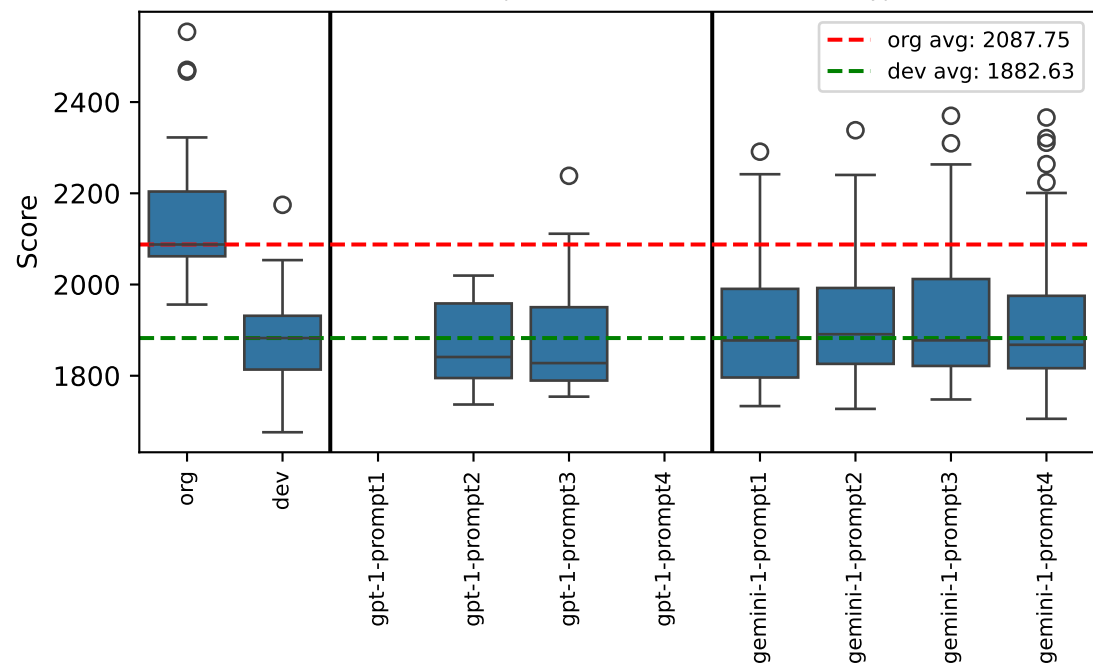
Params: {'numberOfPregroupedColumns': '3', 'partitionsPerGroup': '100', 'rowsPerPartition': '100'}



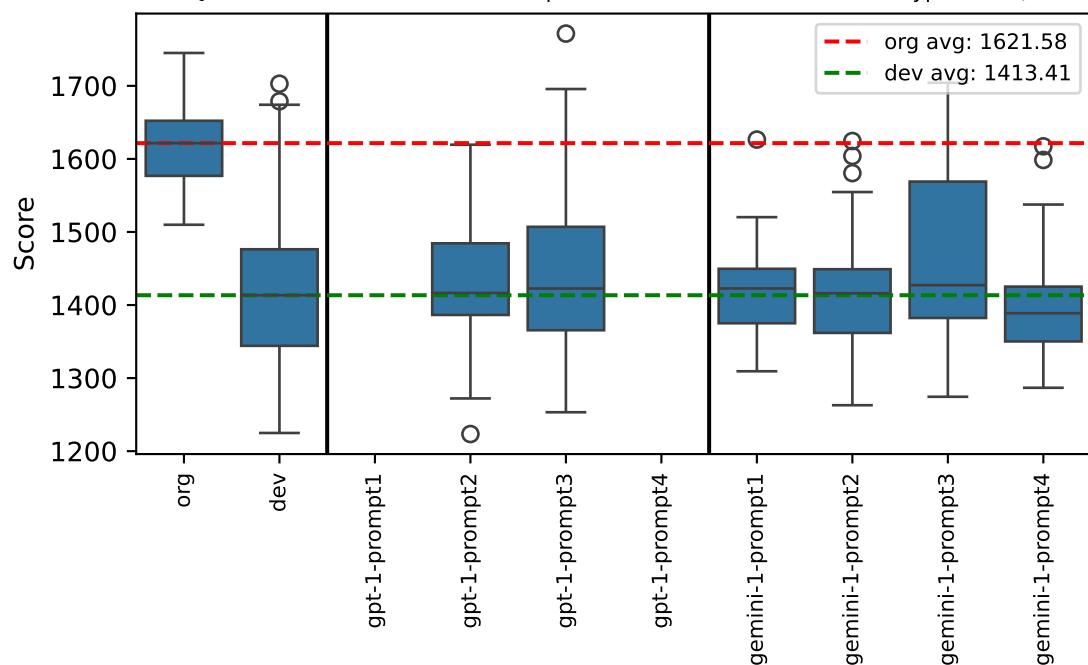
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'BIGINT'}



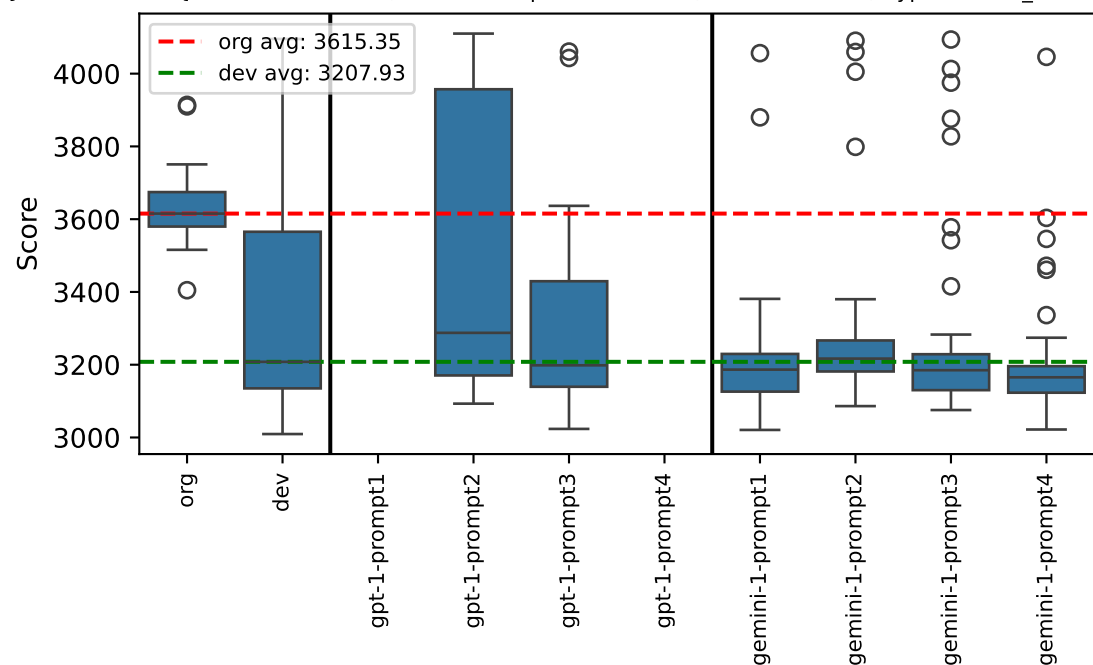
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'DICTIONARY(BIGINT)'}



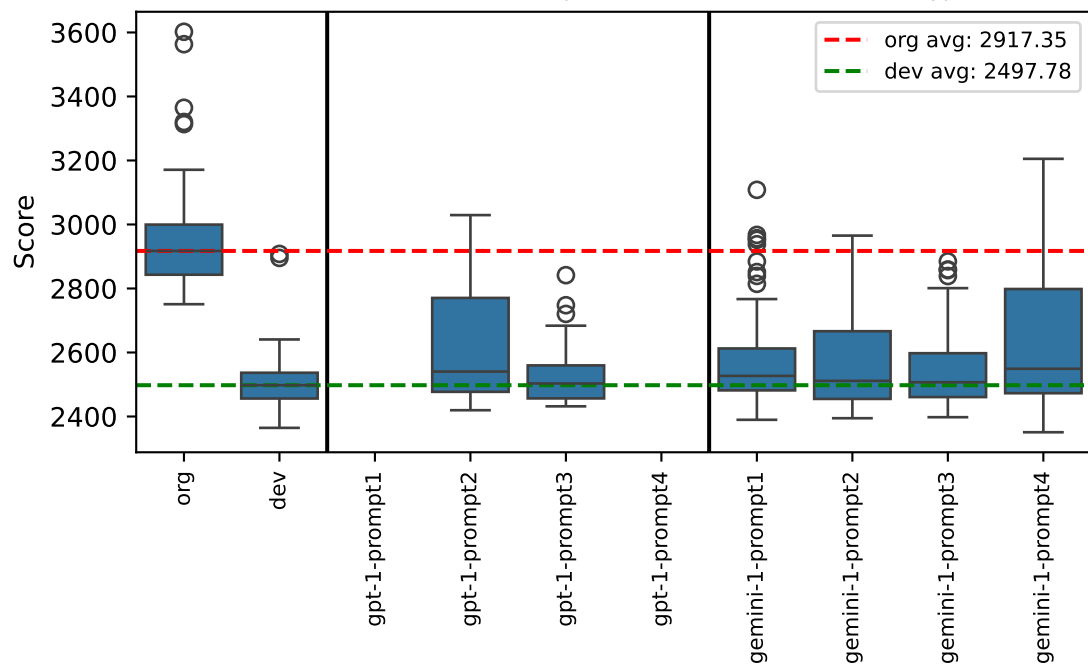
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'RLE(BIGINT)'}



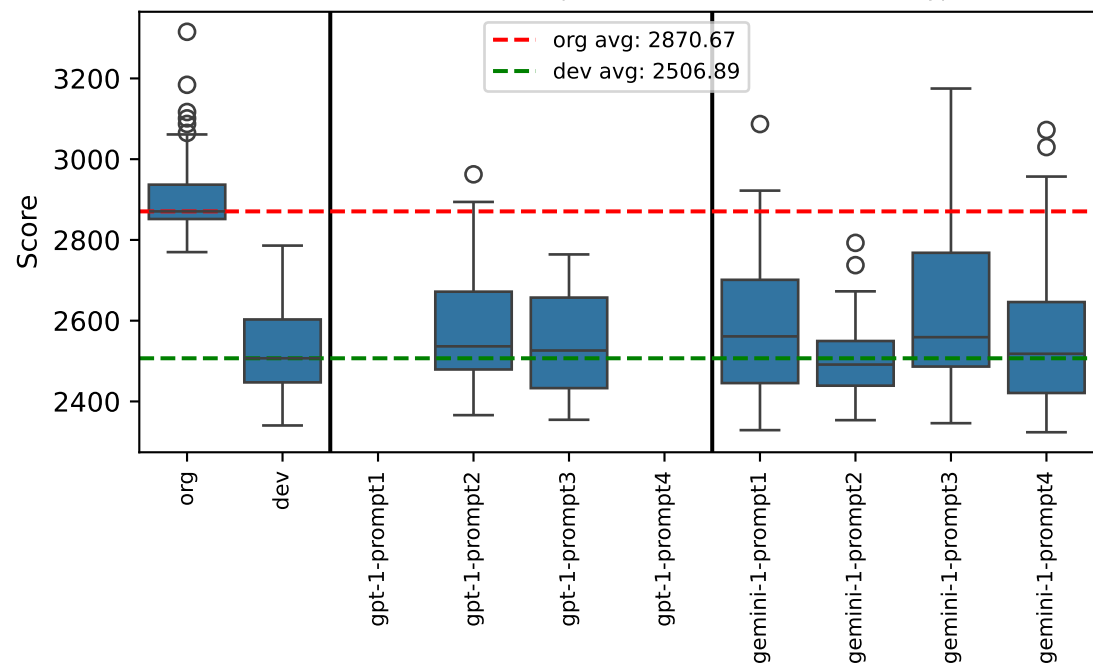
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'LONG_DECIMAL'}



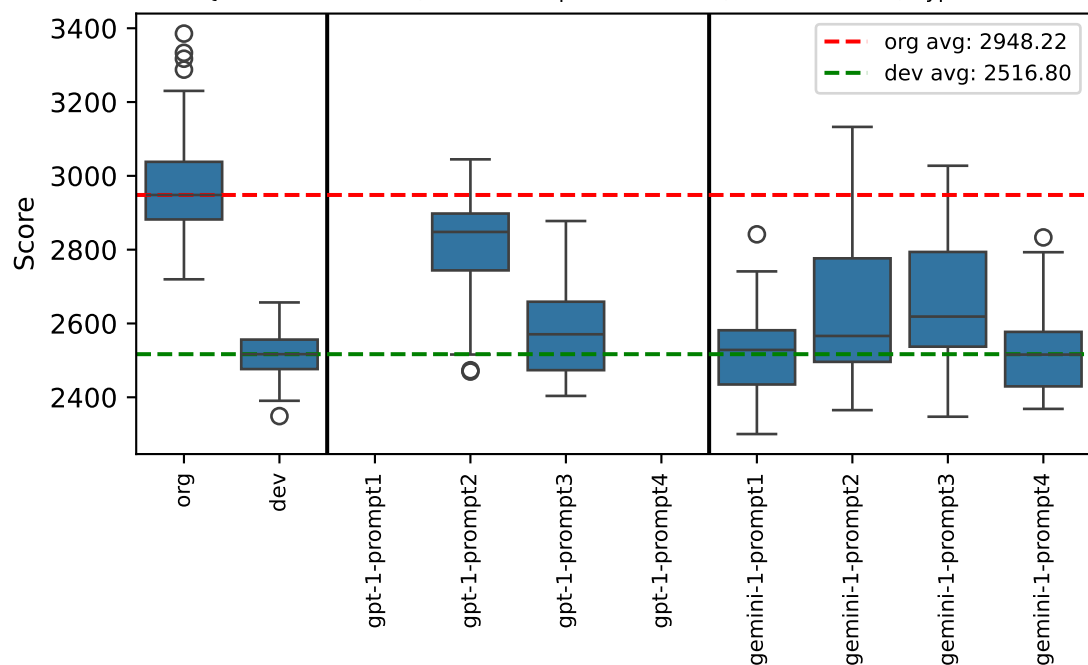
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'INTEGER'}



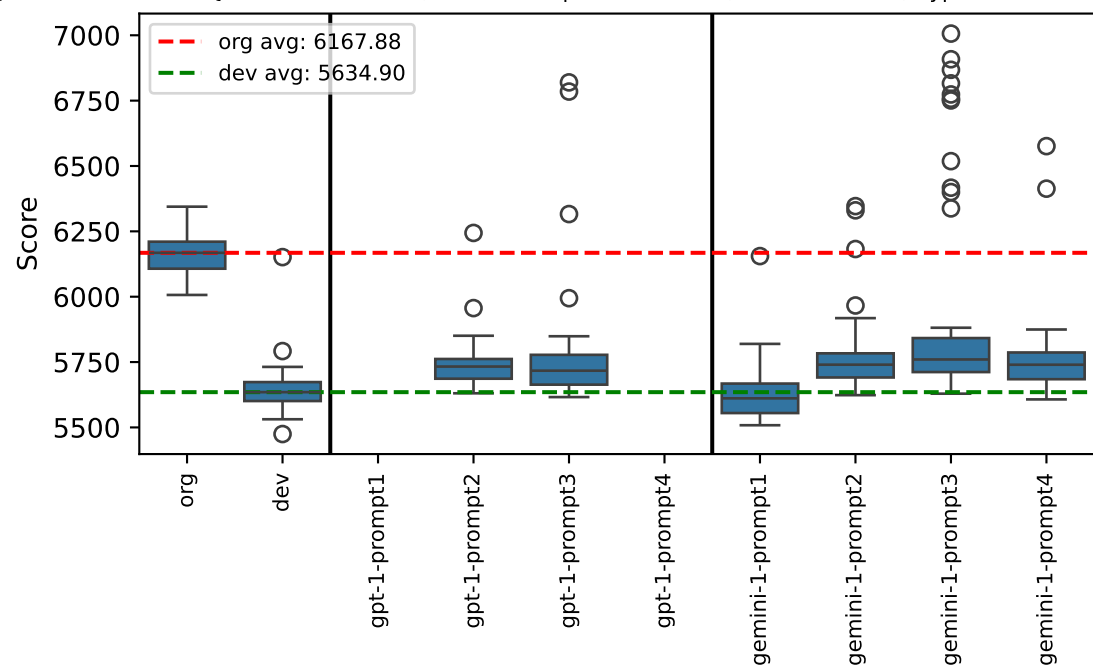
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'SMALLINT'}



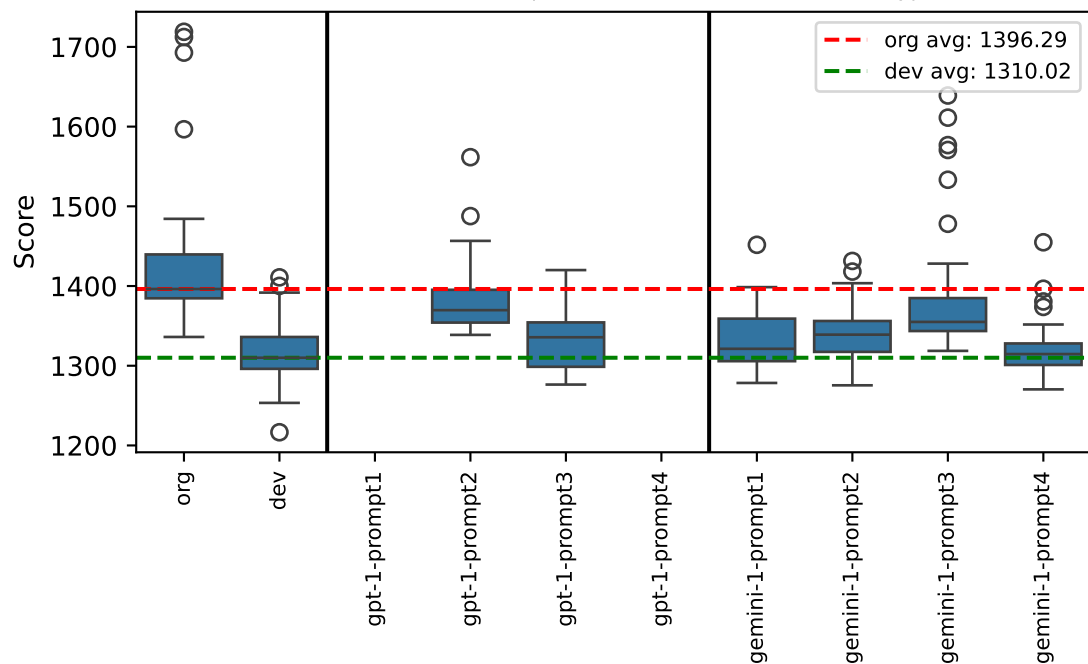
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'BOOLEAN'}



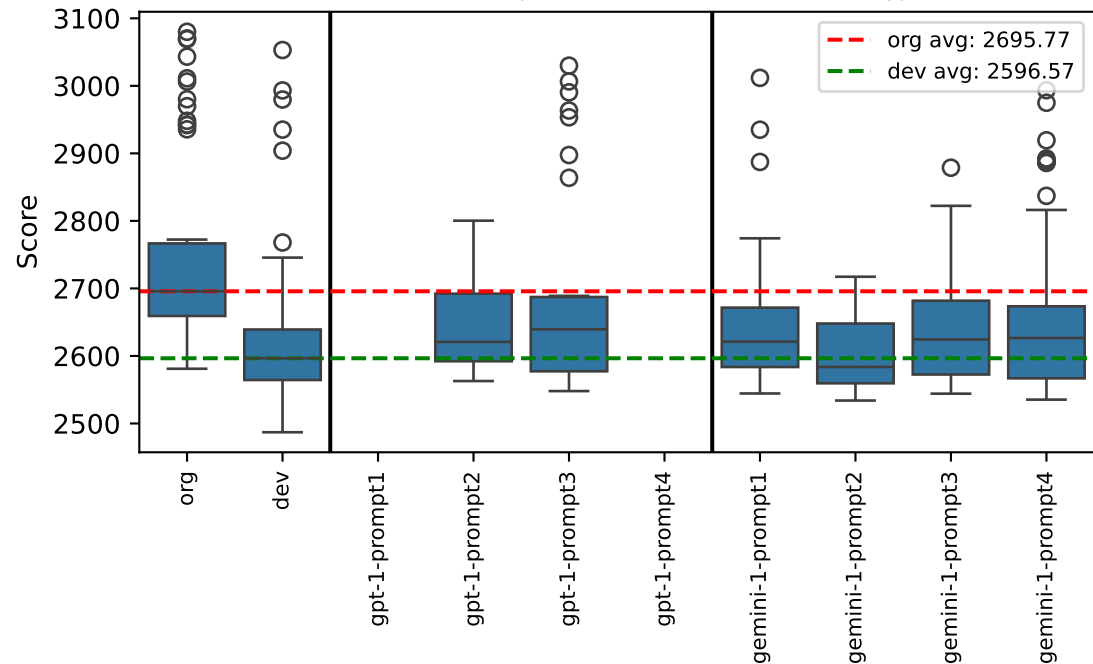
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'VARCHAR'}



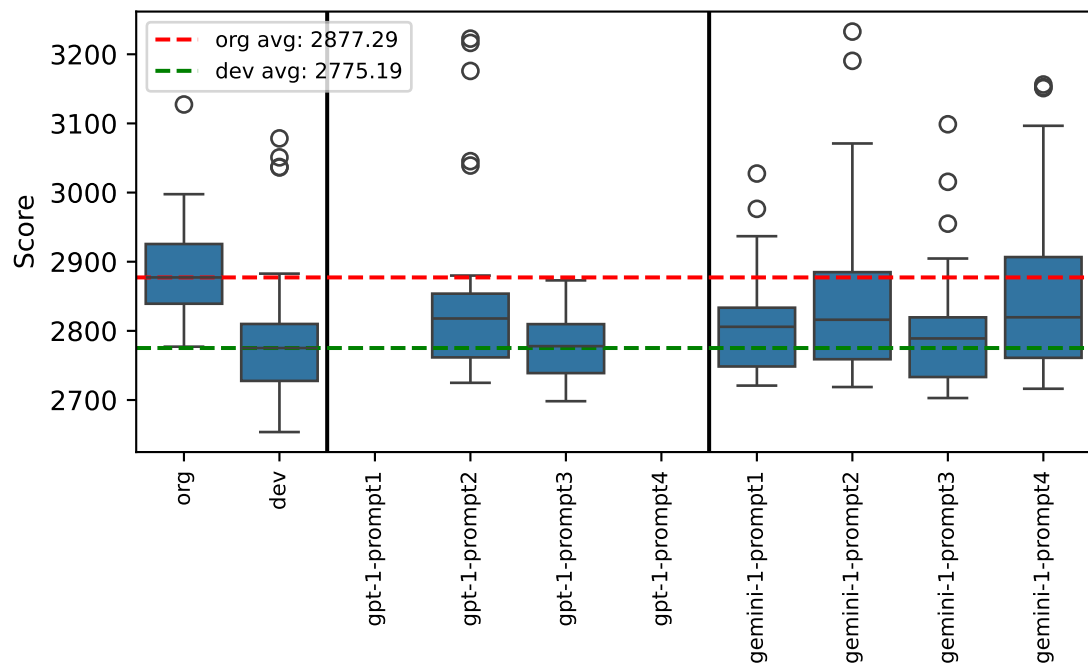
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(BIGINT)'}



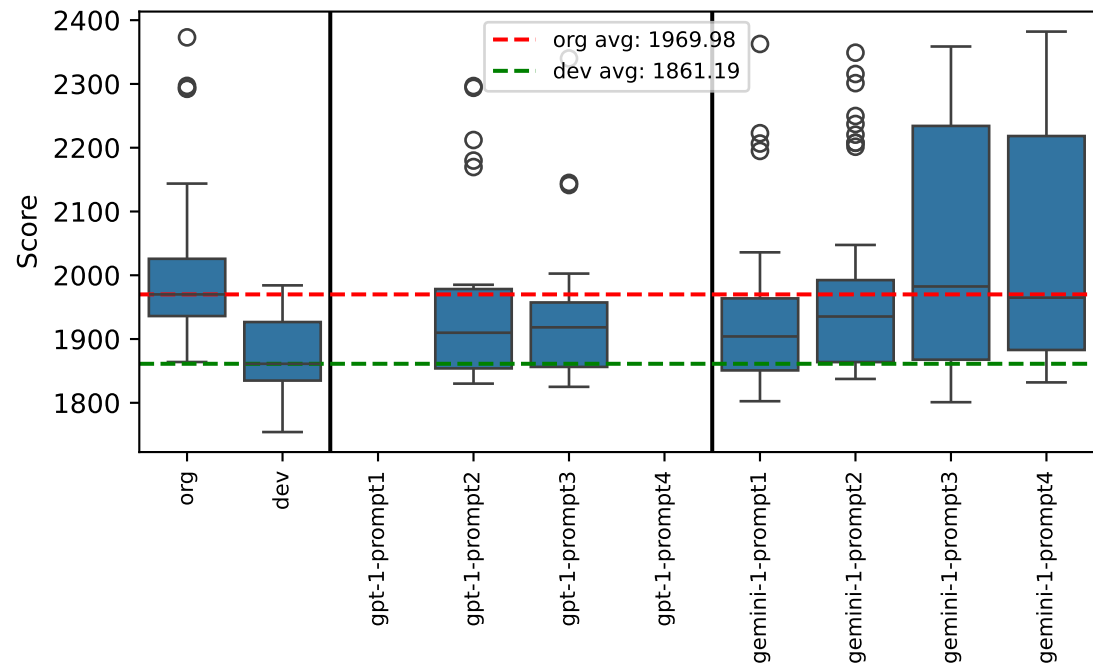
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(VARCHAR)'}



Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(ARRAY(BIGINT))'}

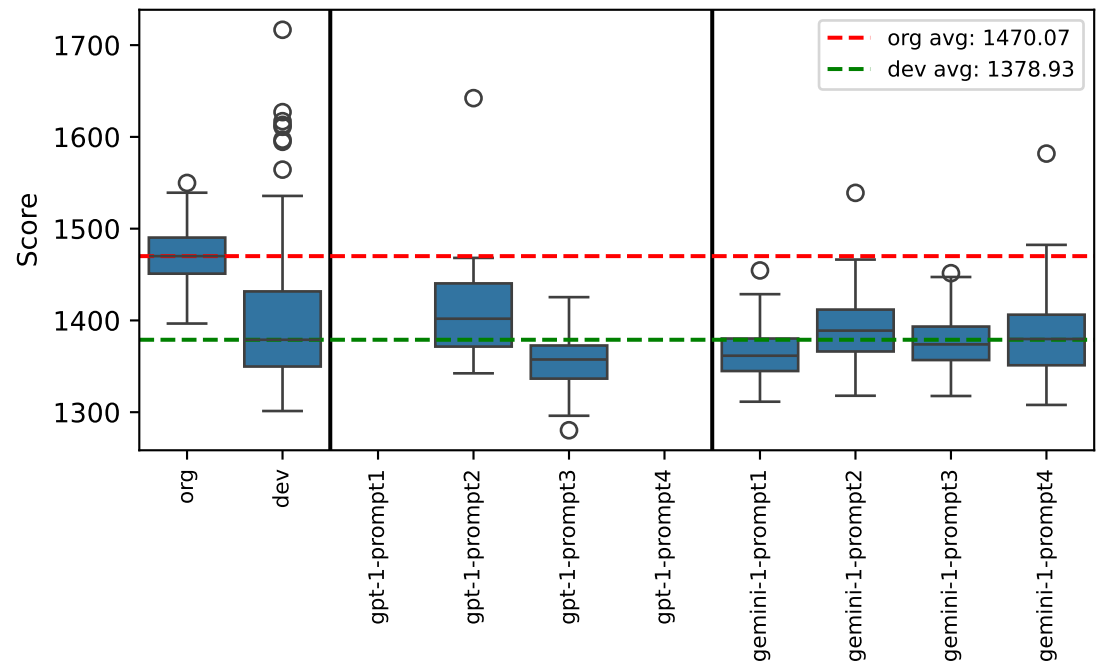
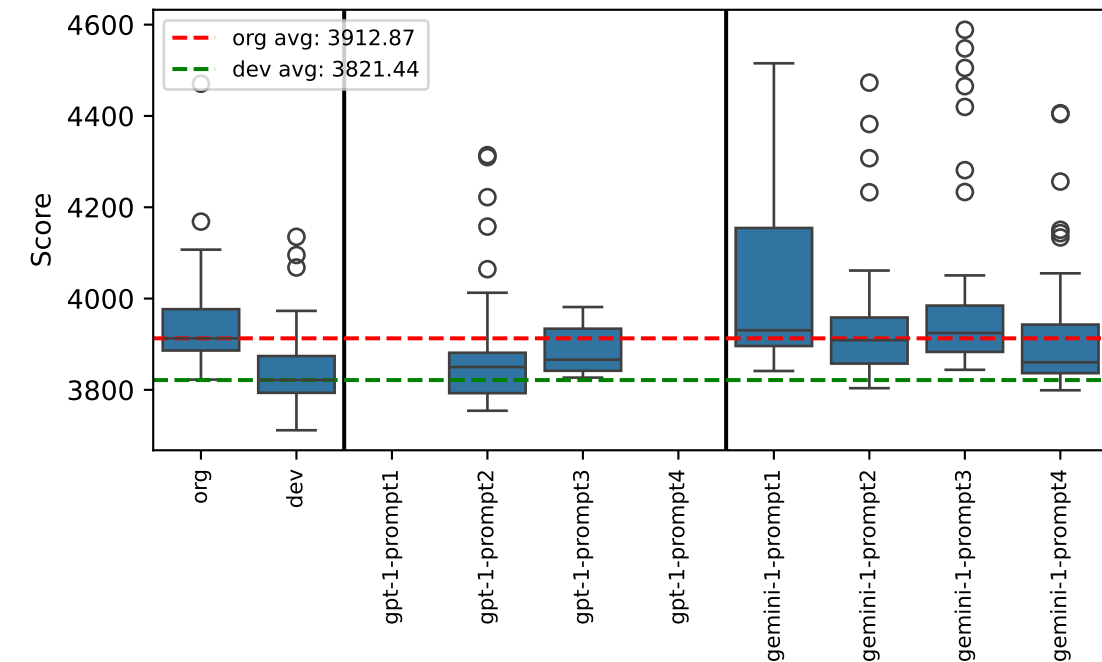


Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'MAP(BIGINT, BIGINT)'}

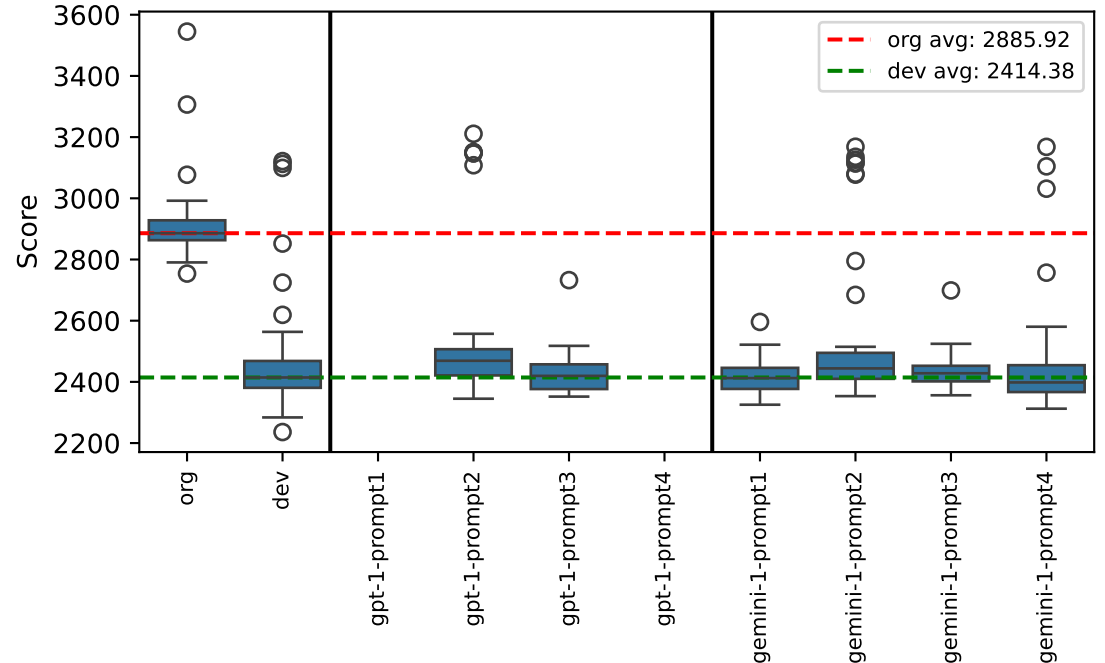
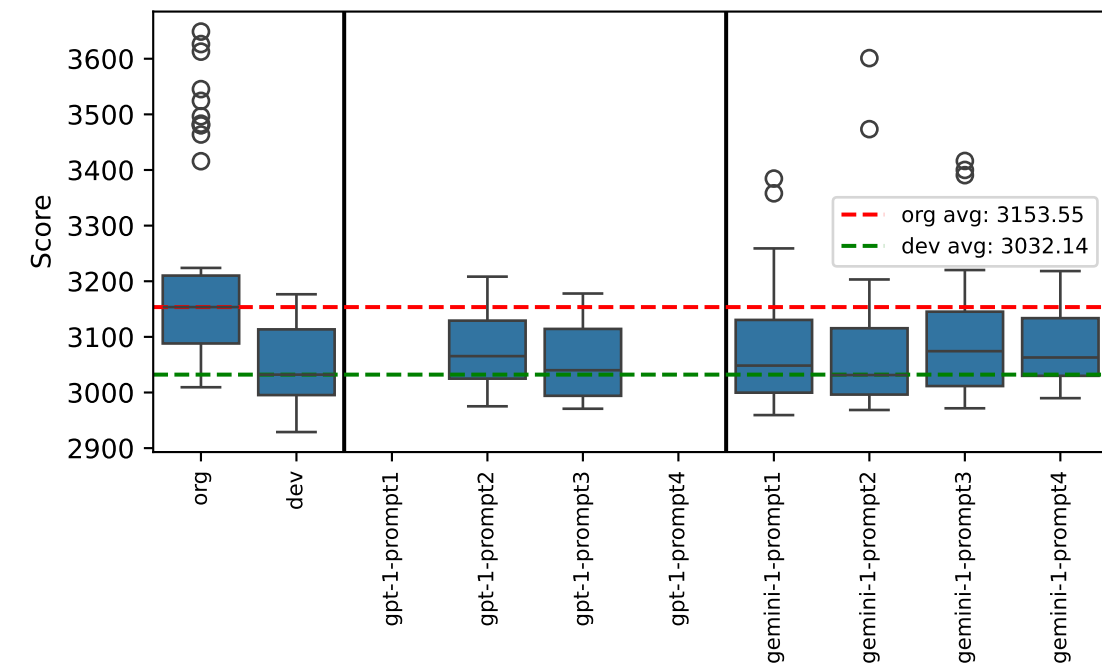


Benchmark: com.facebook.presto.operator.repartition.BenchmarkPartitionedOutputOperator.optimizedAddPa

Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'MAP(BIGINT, MAP(BIGINT, BIGINT))'} Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ROW(BIGINT, BIGINT)}

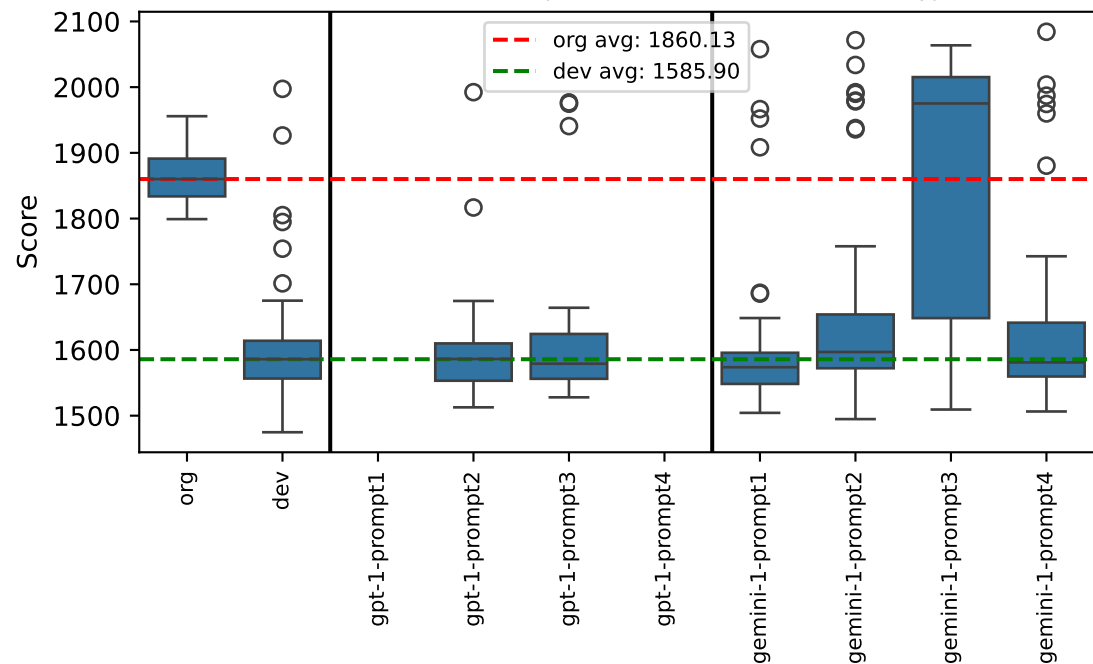
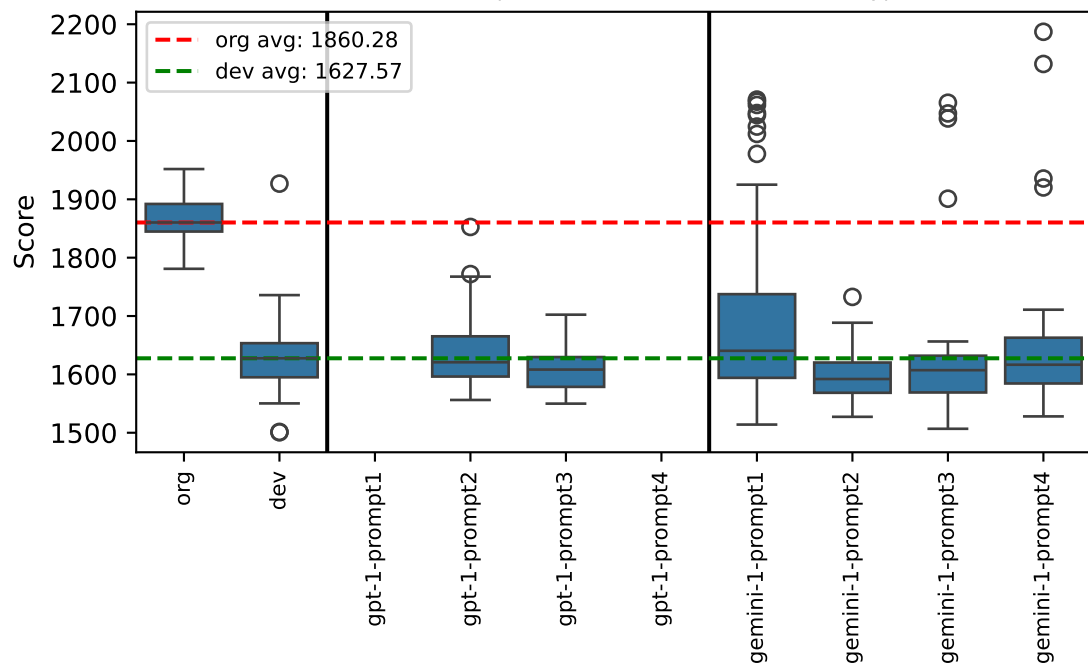


Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'} Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'BIGINT'}



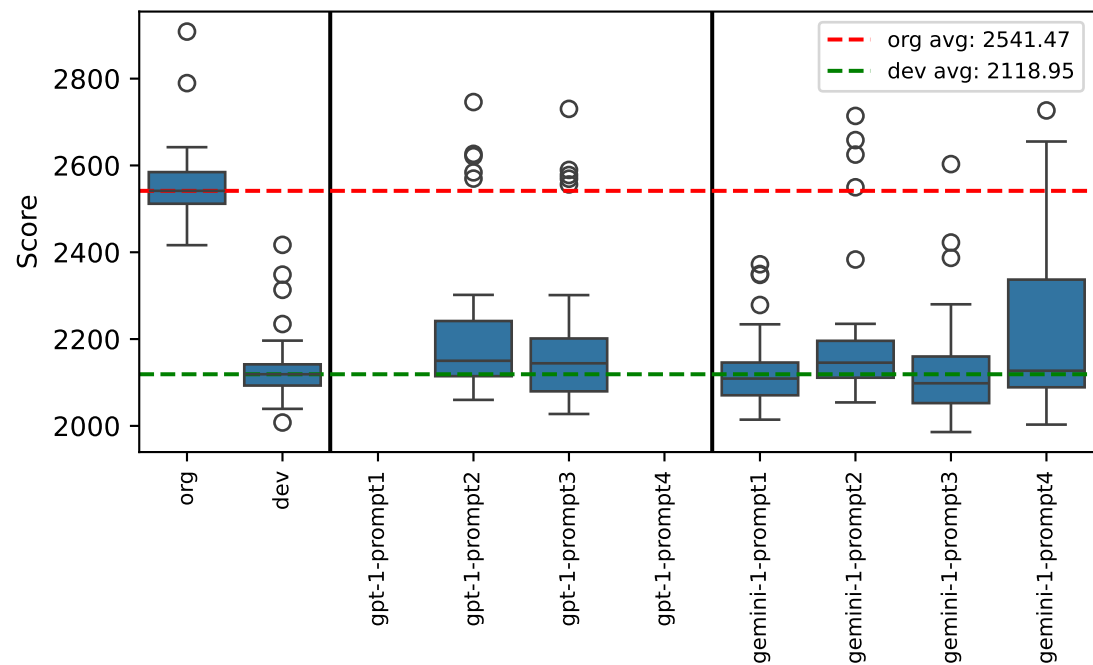
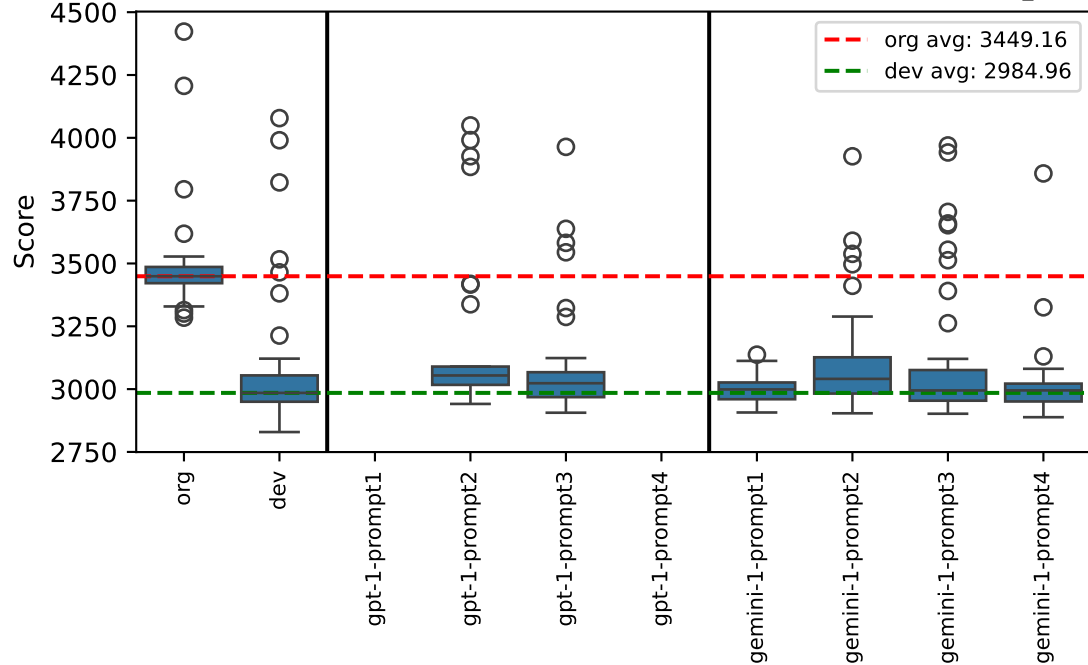
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'DICTIONARY(BIGINT)'}

Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'RLE(BIGINT)'}

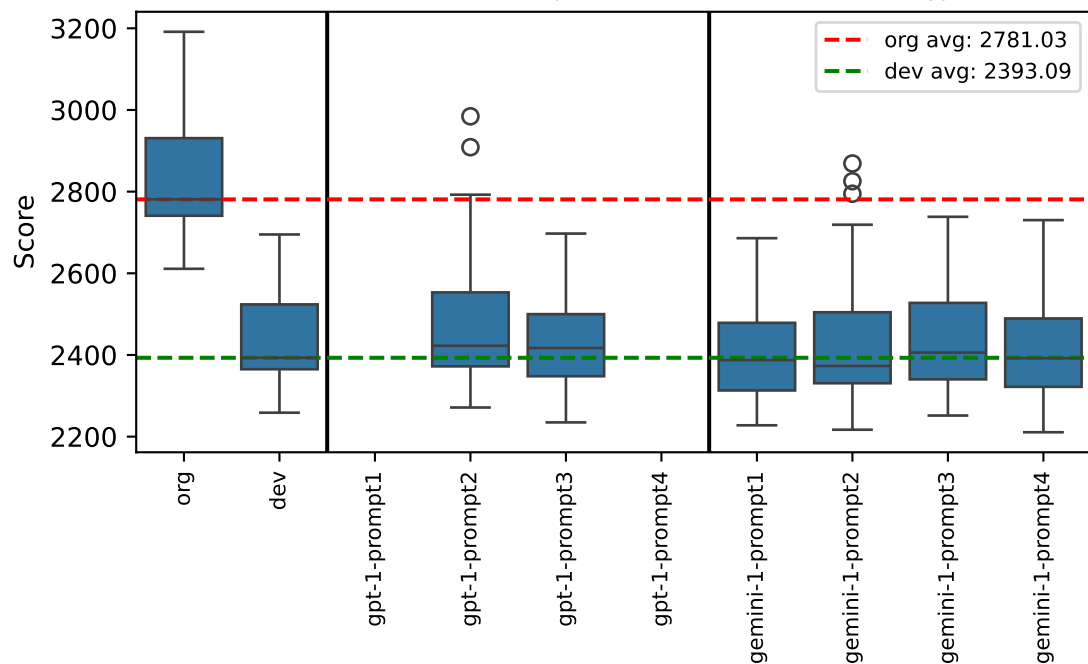


Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'LONG_DECIMAL'}

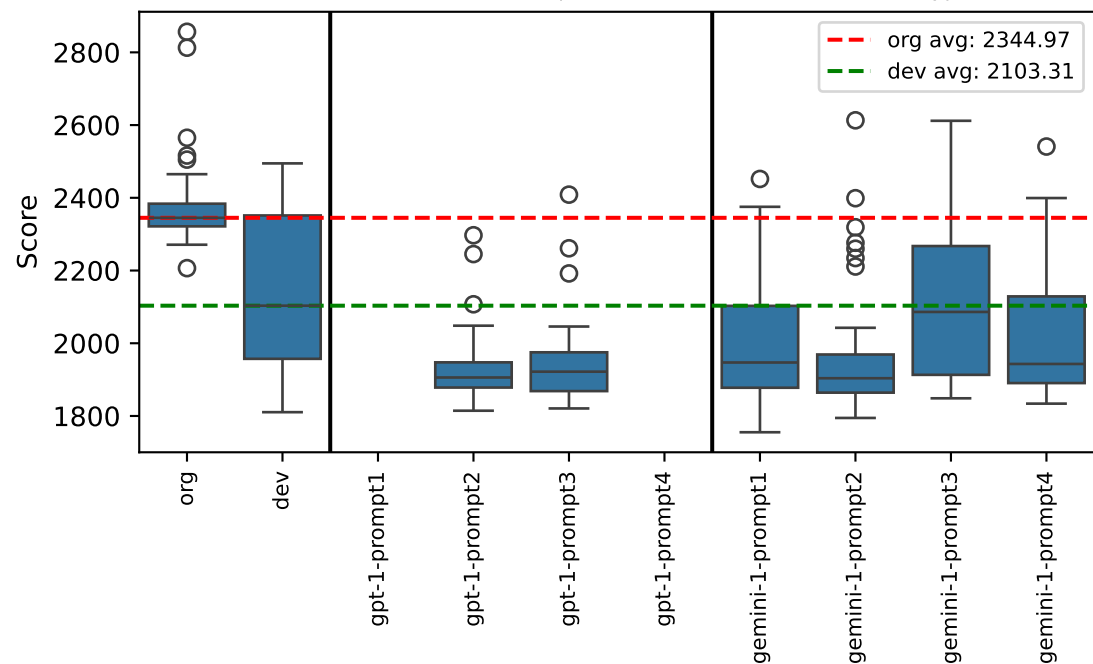
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'INTEGER'}



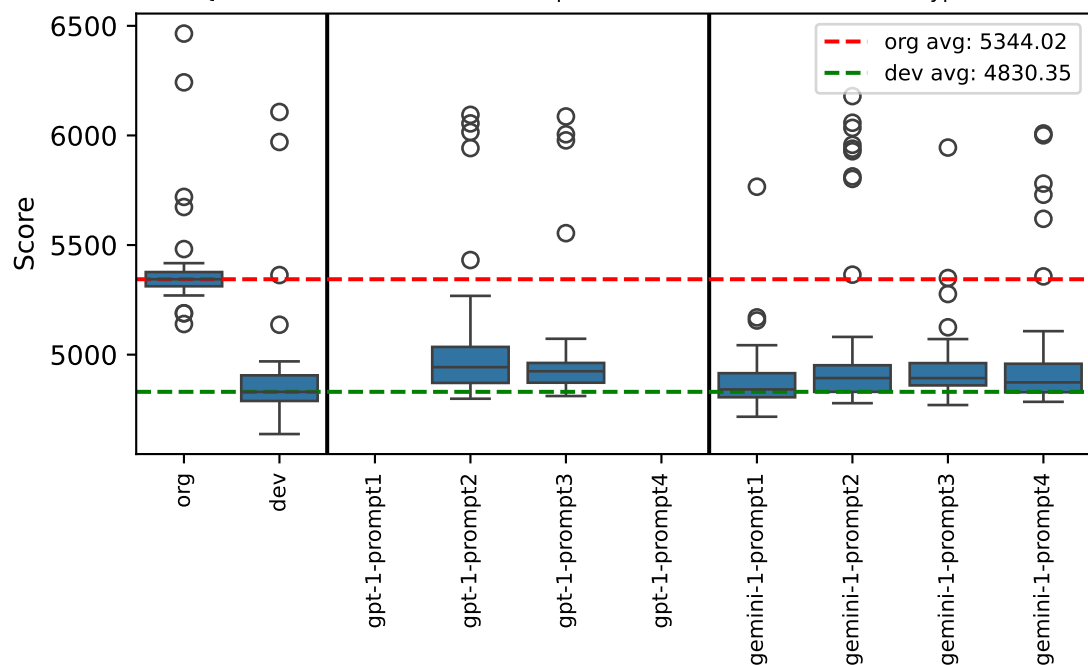
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'SMALLINT'}



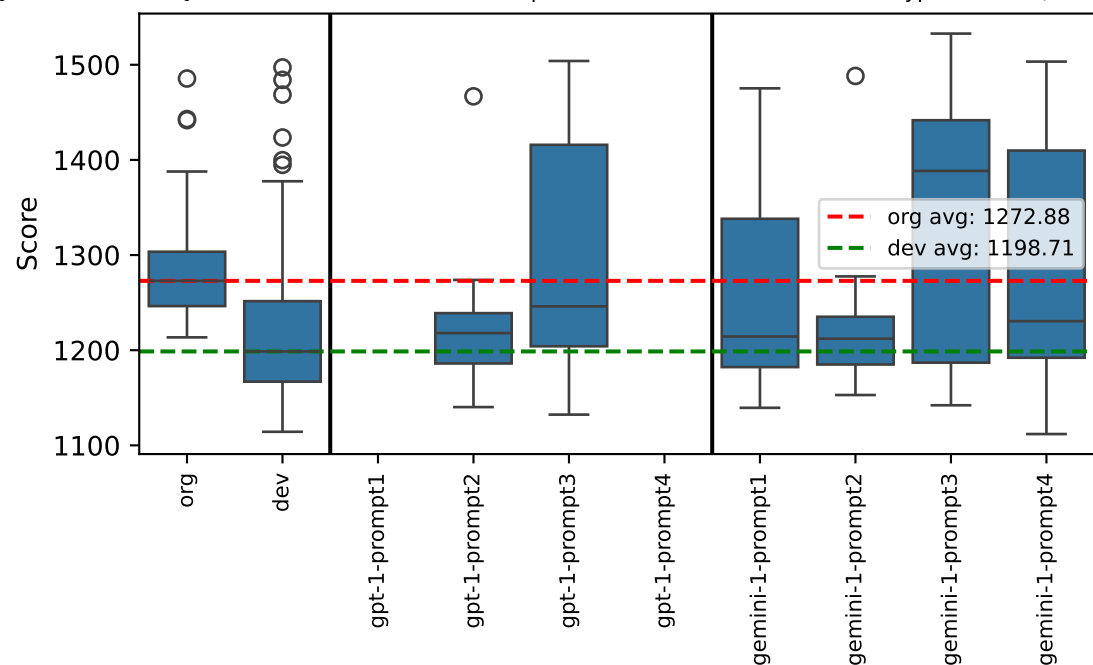
Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'BOOLEAN'}

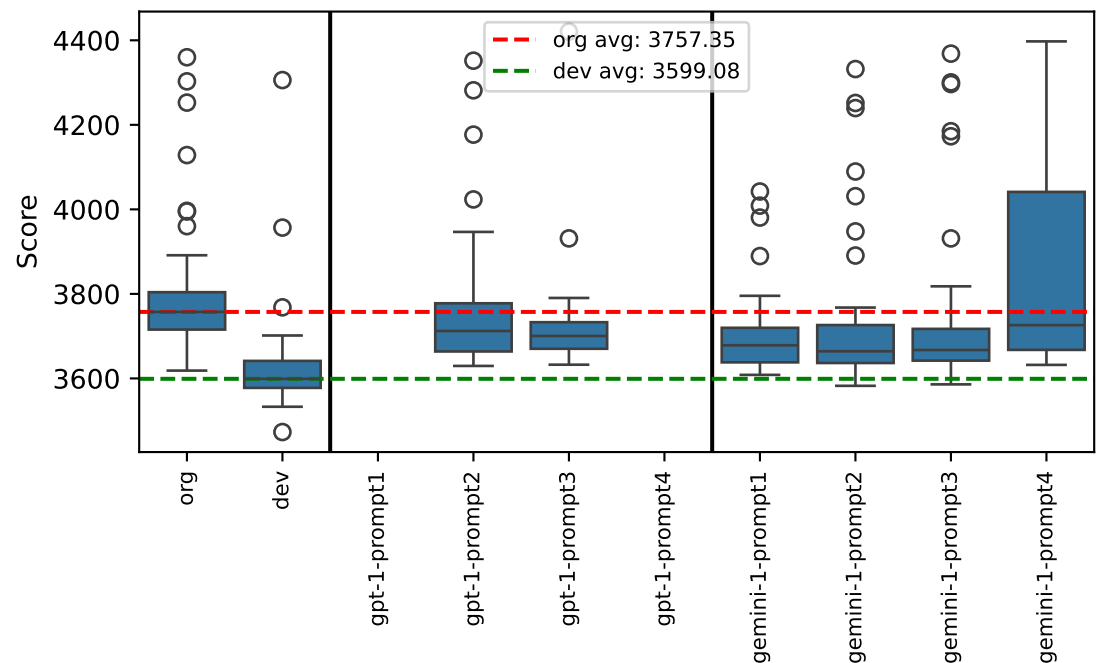
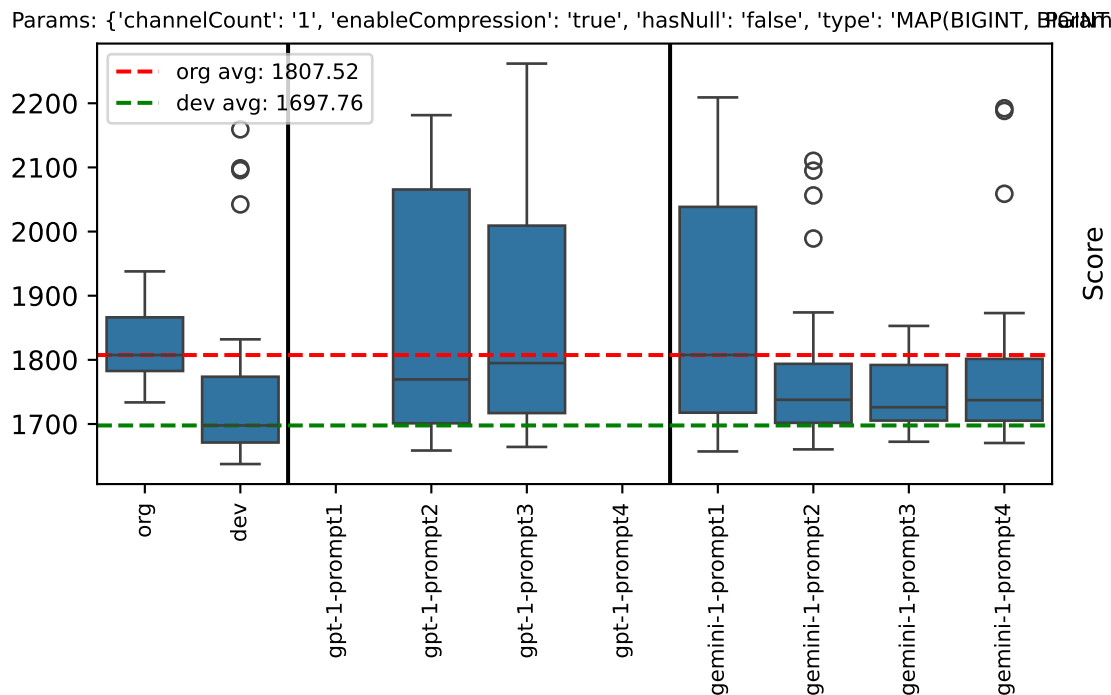
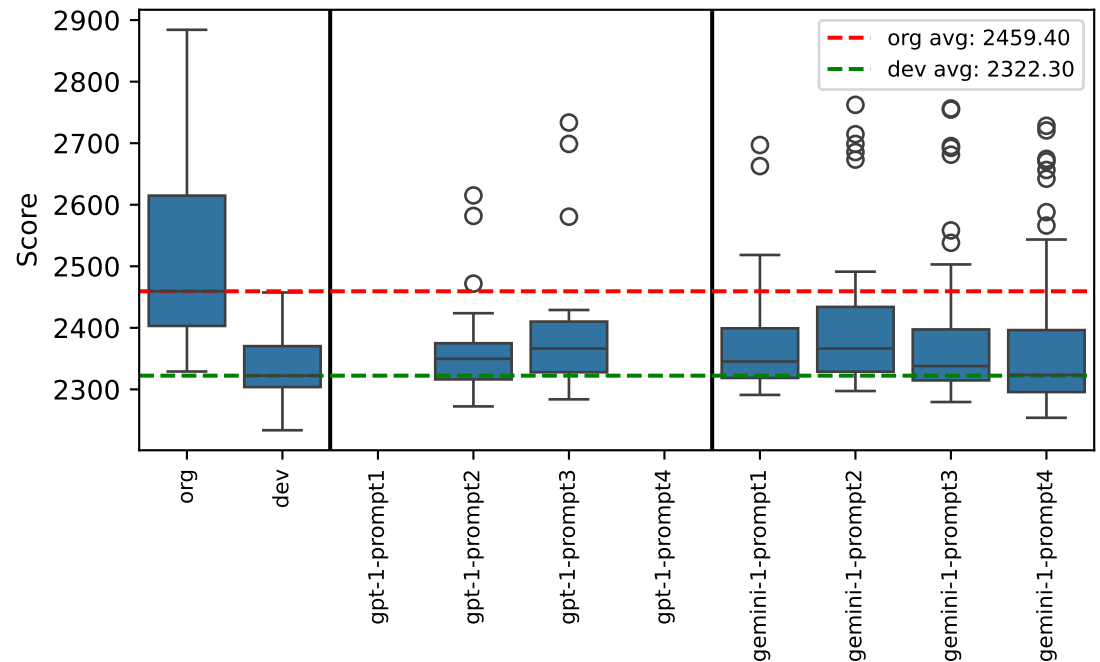
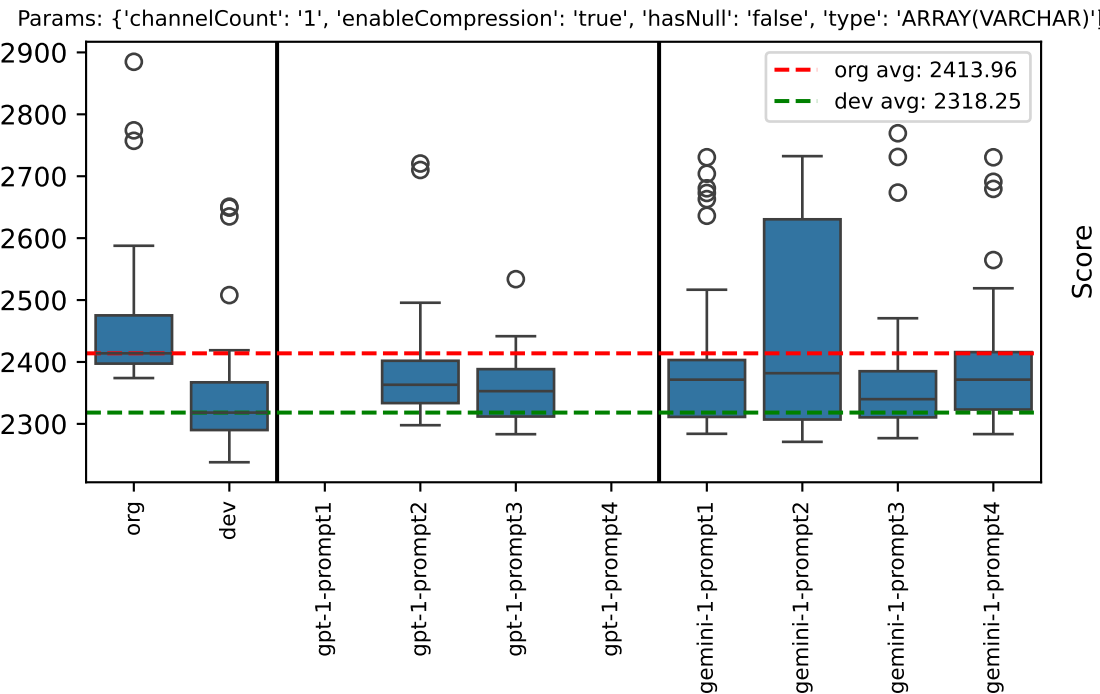


Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'VARCHAR'}

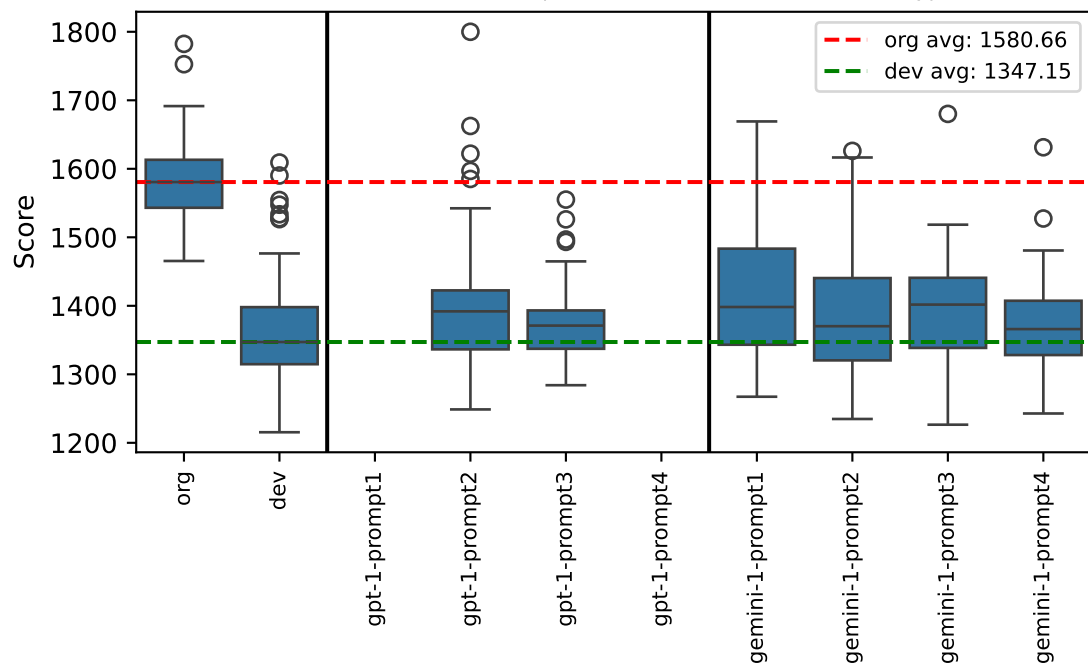


Params: {'channelCount': '1', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'ARRAY(BIGINT)'}

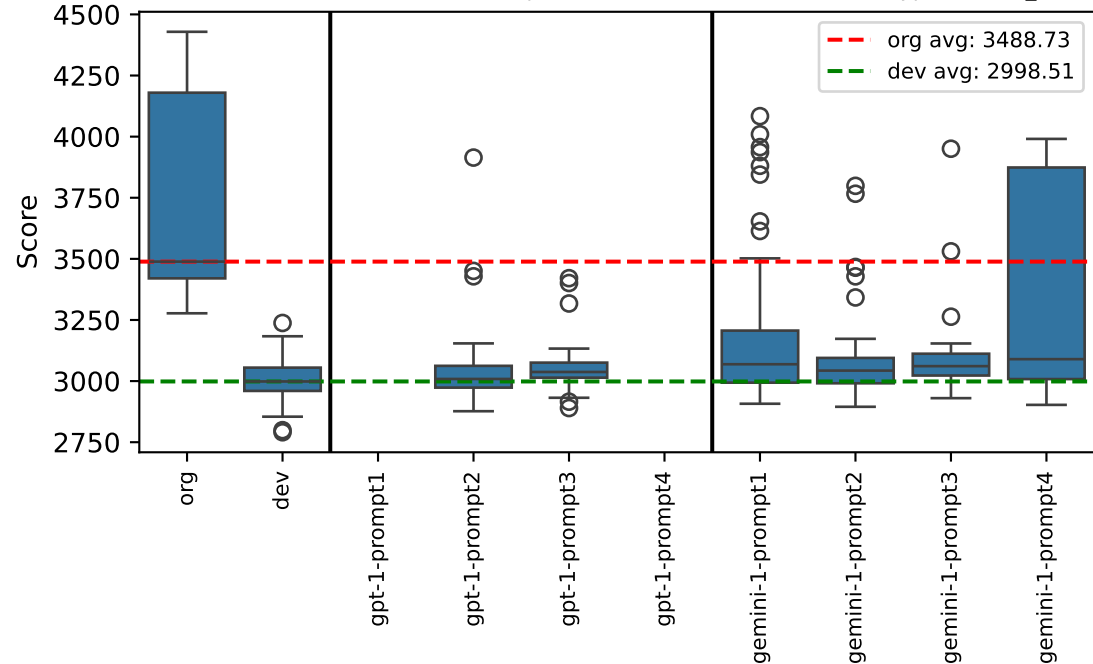




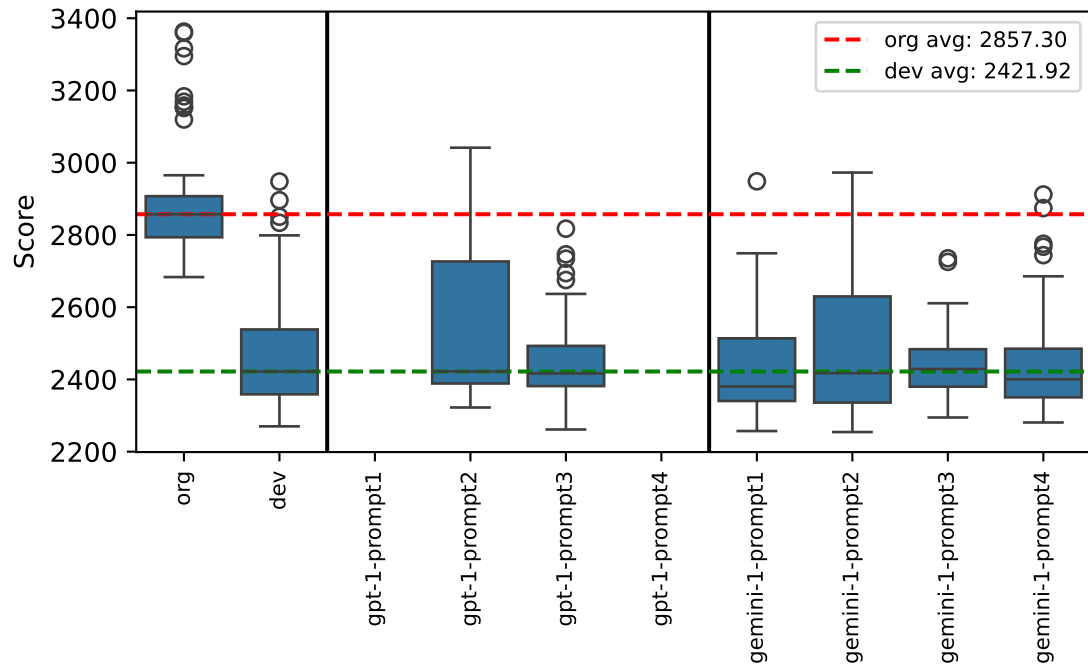
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'RLE(BIGINT)'}



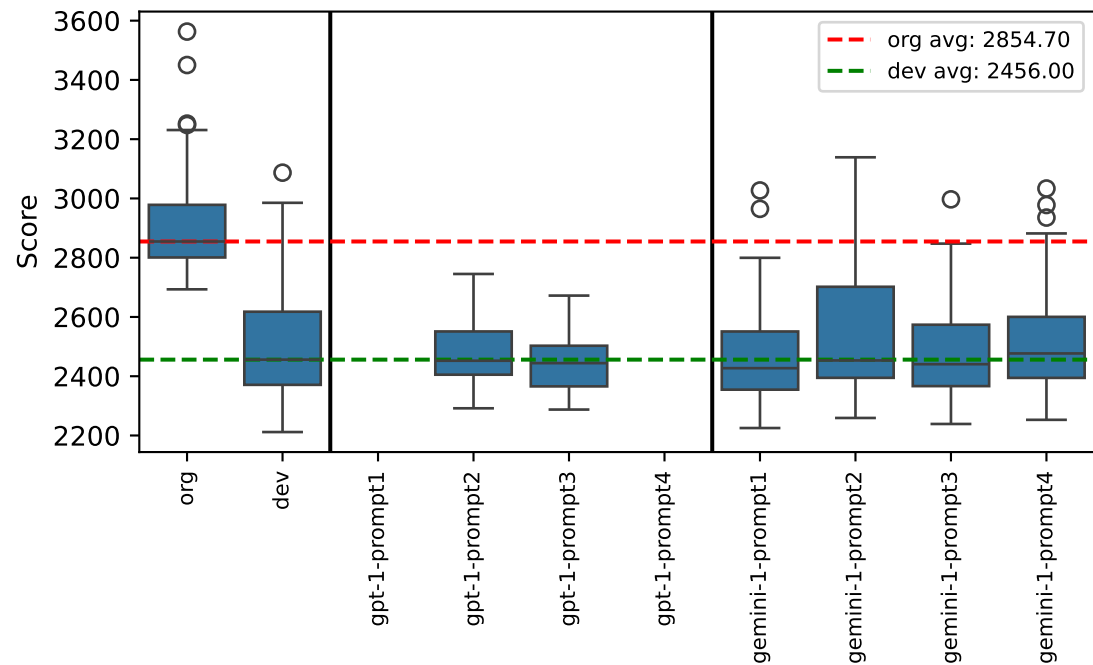
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'LONG_DECIMAL'



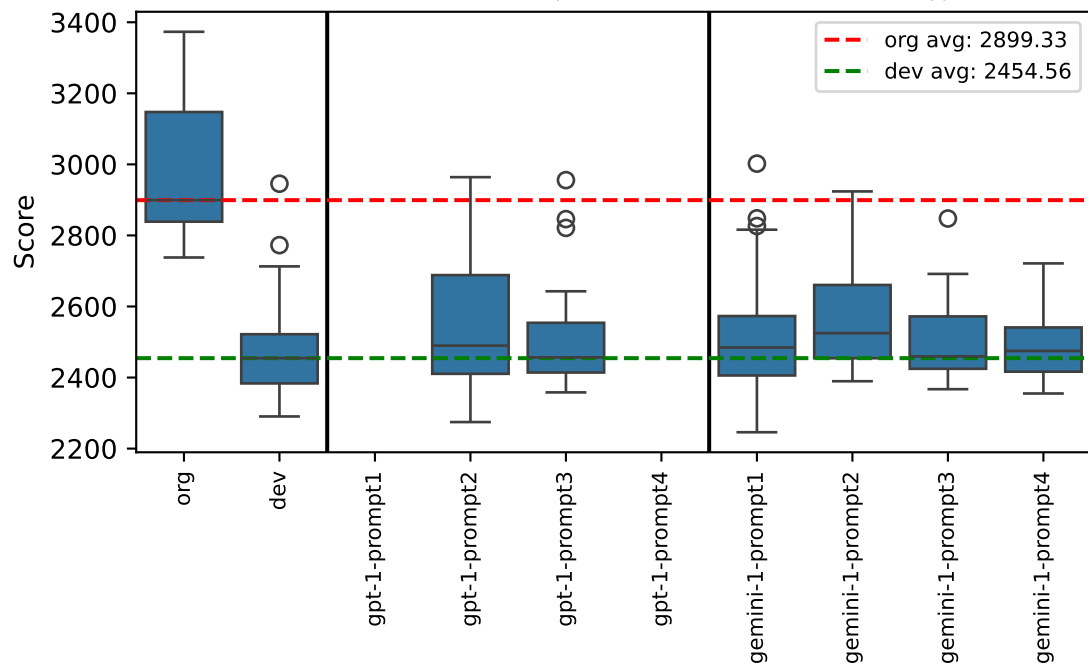
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'INTEGER'}



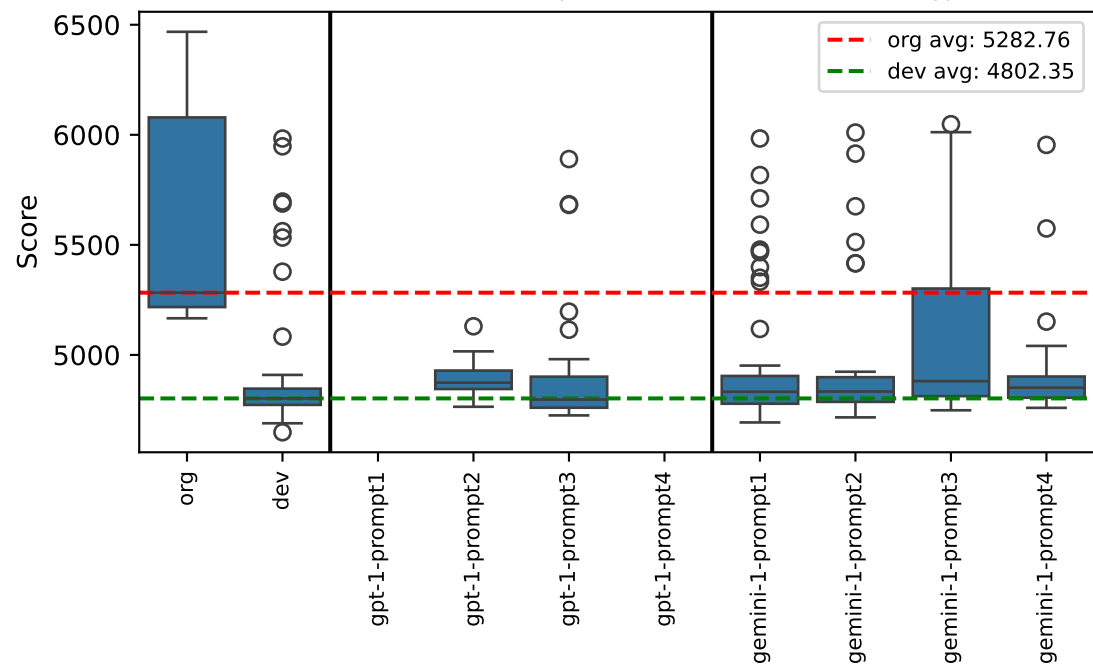
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'SMALLINT'}



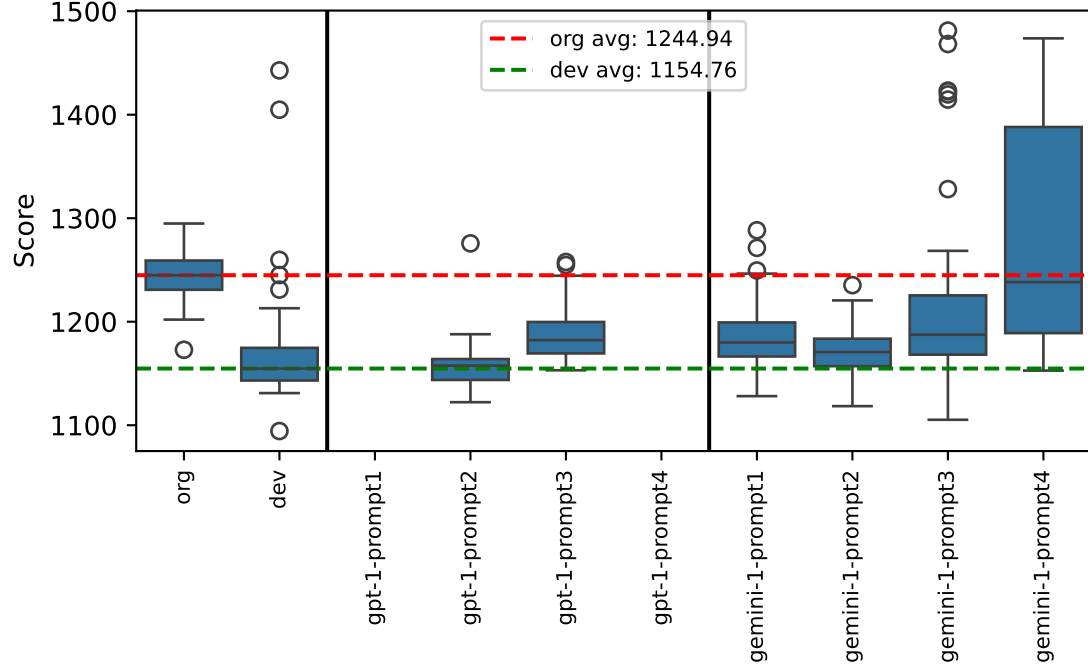
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'BOOLEAN'}



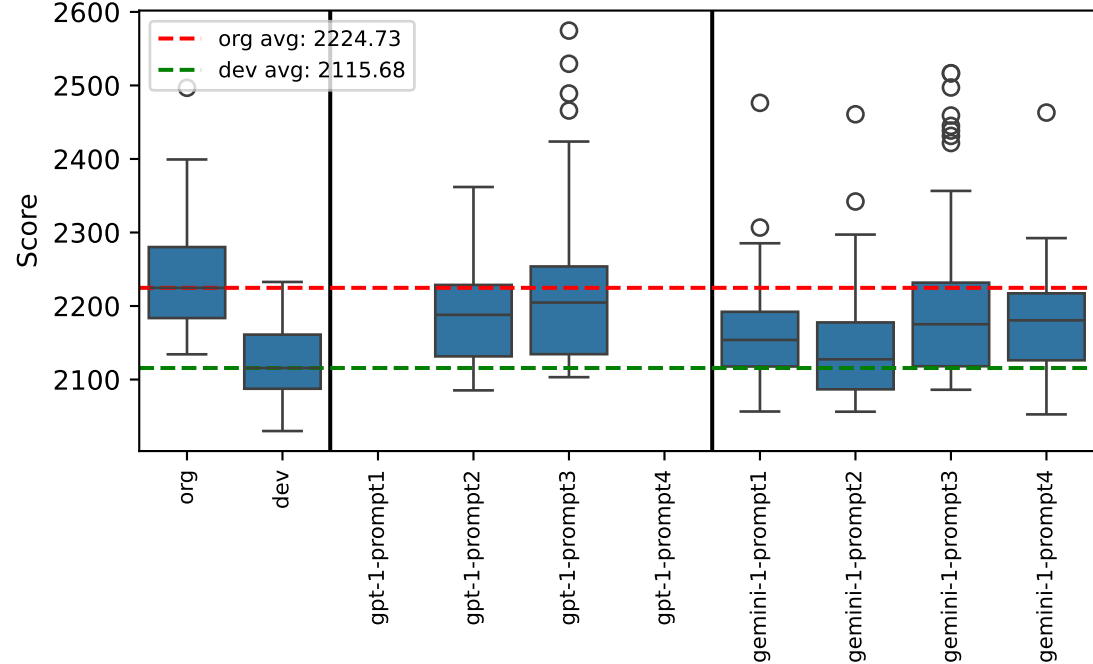
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'VARCHAR'}

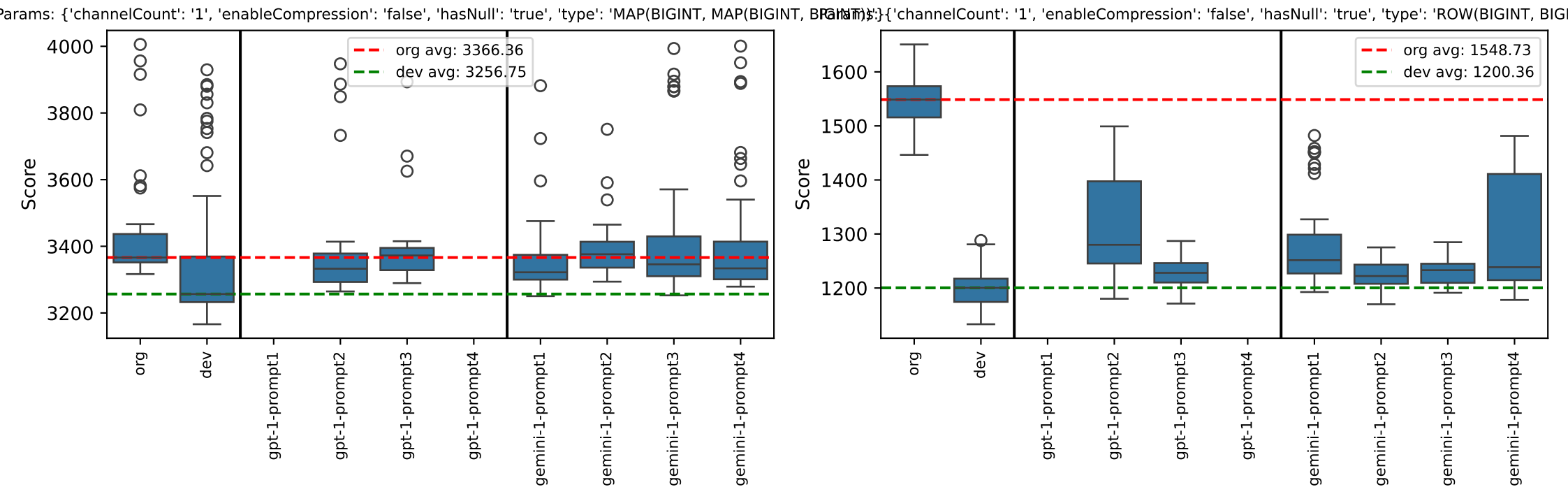
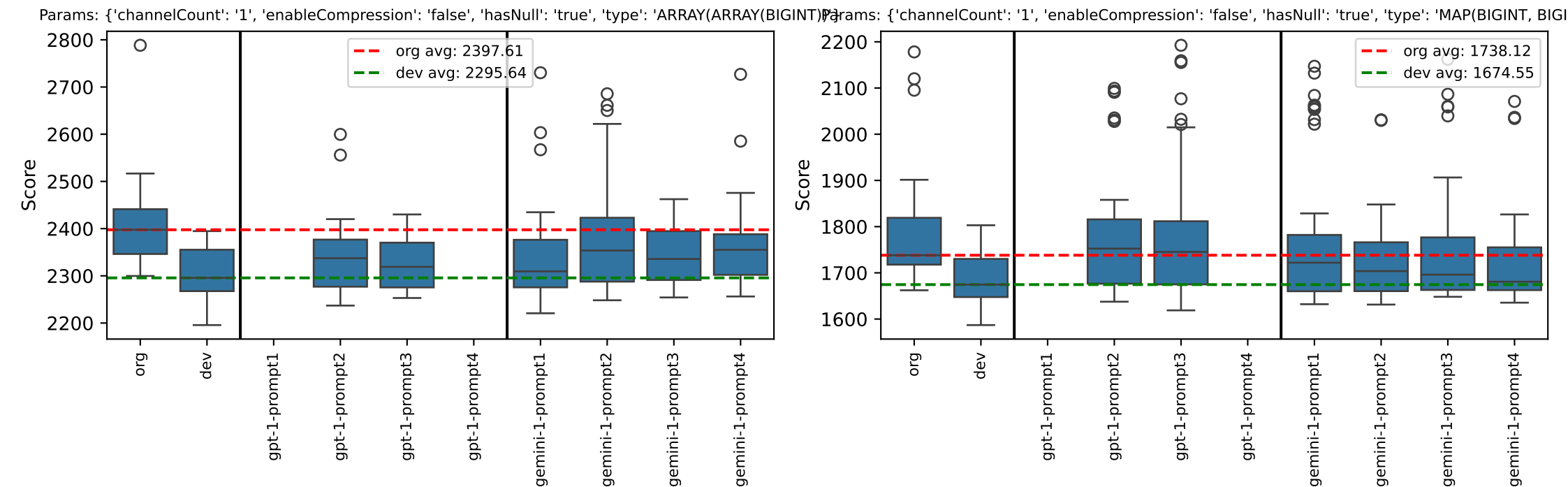


Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ARRAY(BIGINT)'}



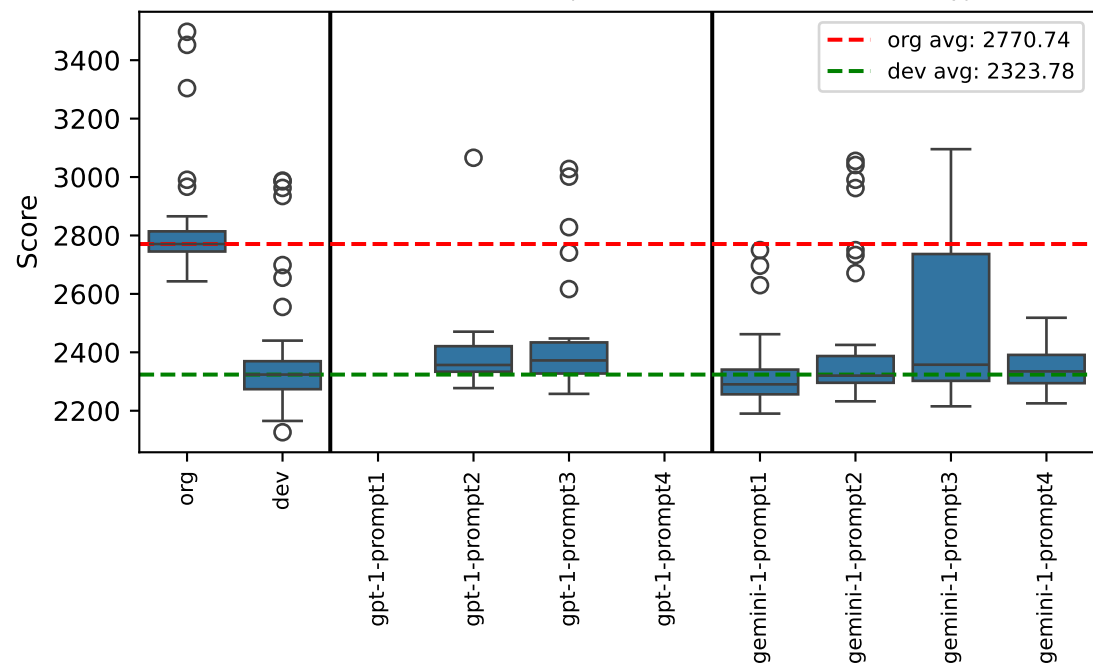
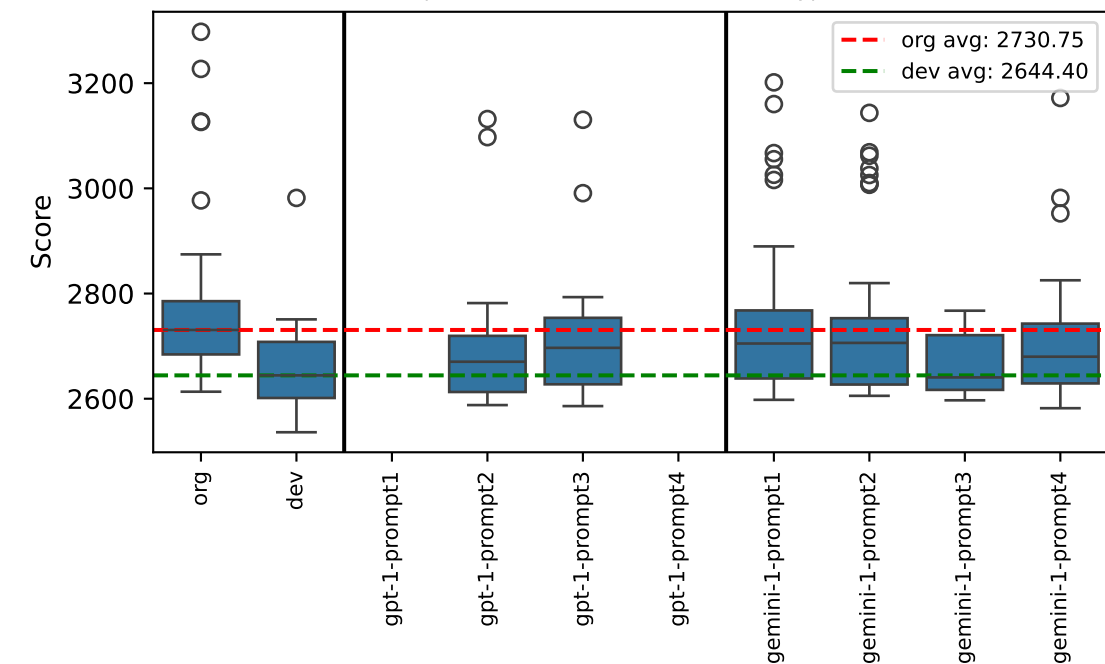
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ARRAY(VARCHAR)'}



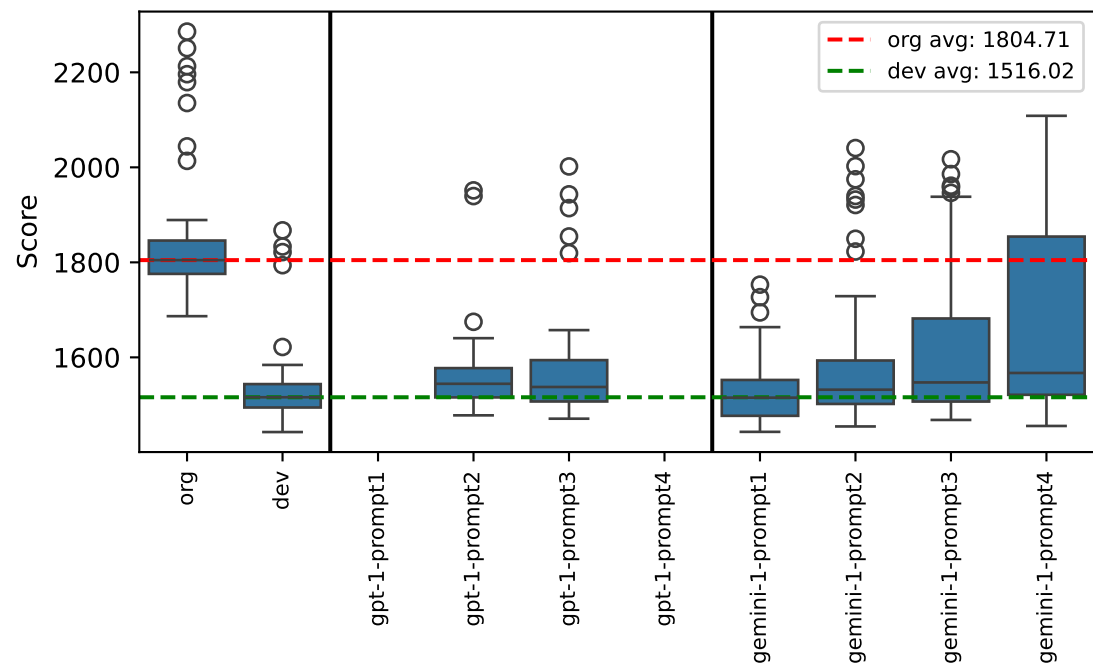
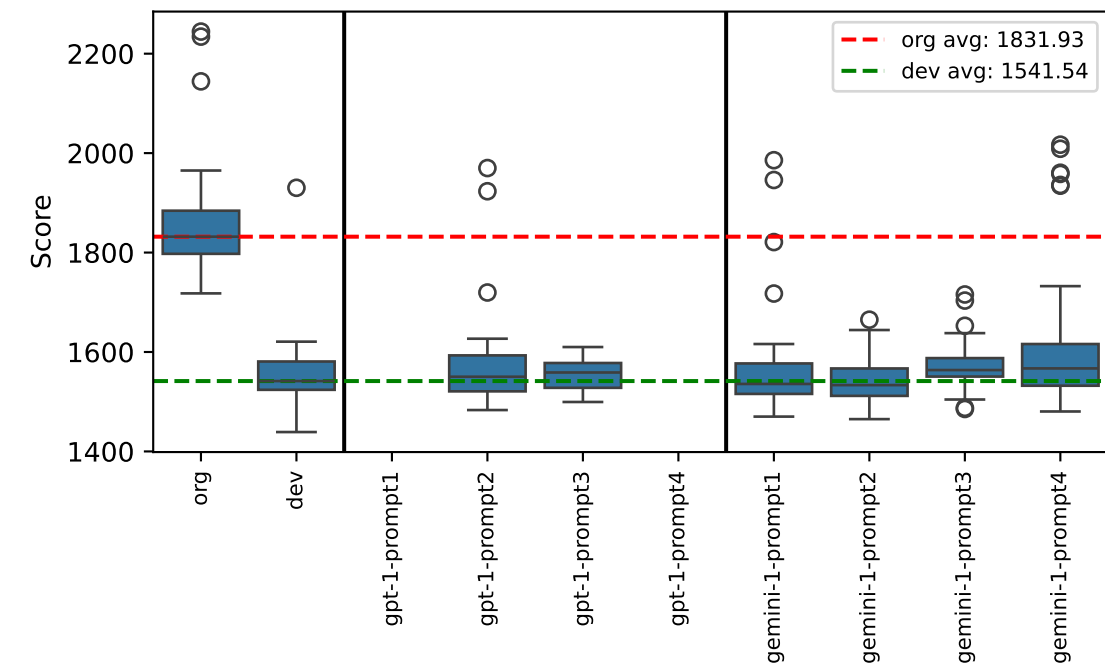


Benchmark: com.facebook.presto.operator.repartition.PartitionedOutputOperator.optimizedAddPa

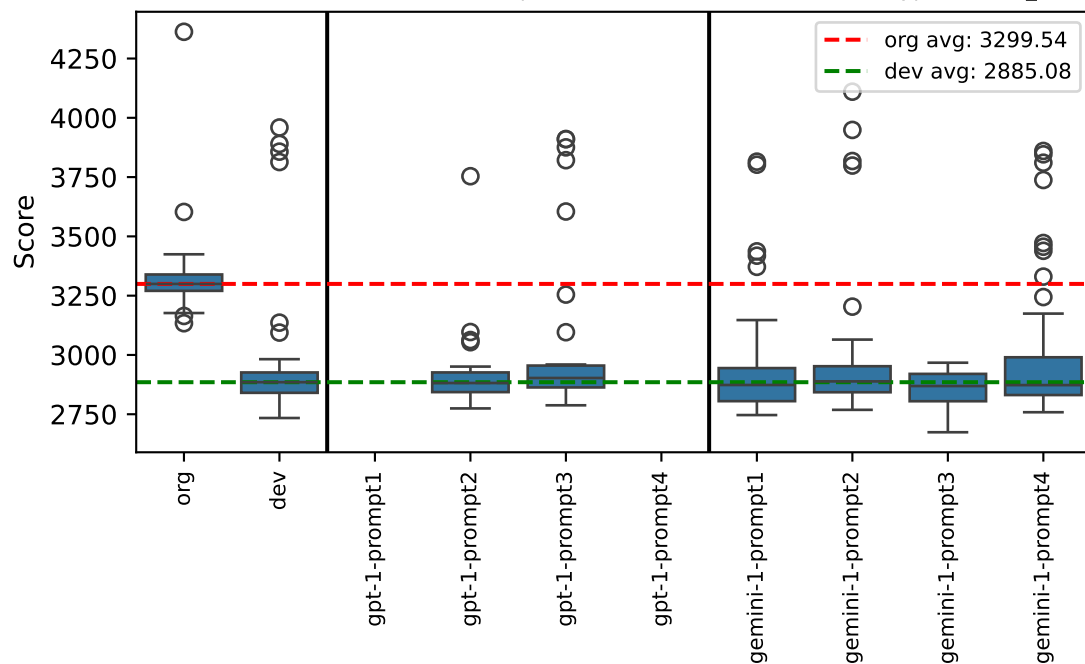
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'} Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'BIGINT'}



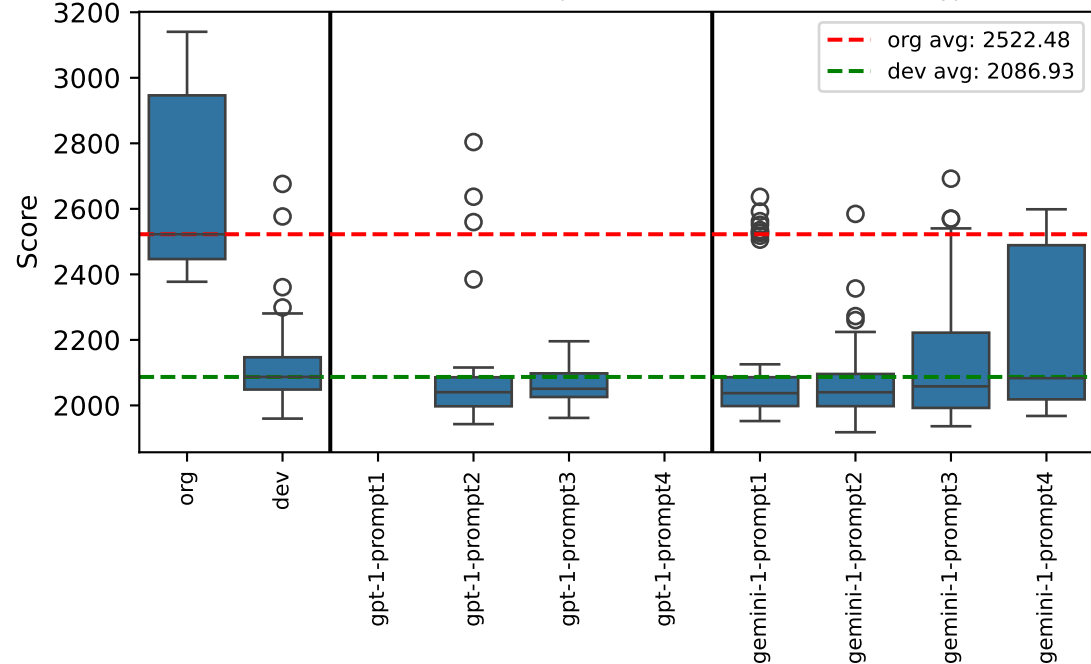
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'DICTIONARY(BIGINT)'} Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'RLE(BIGINT)'}



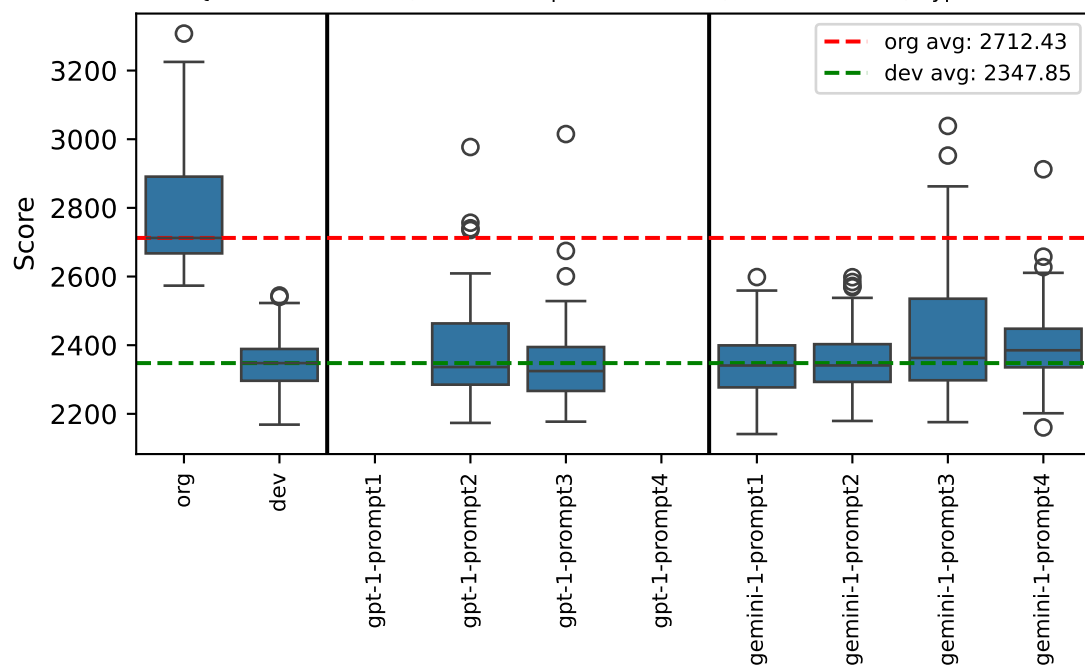
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'LONG_DECIMAL'}



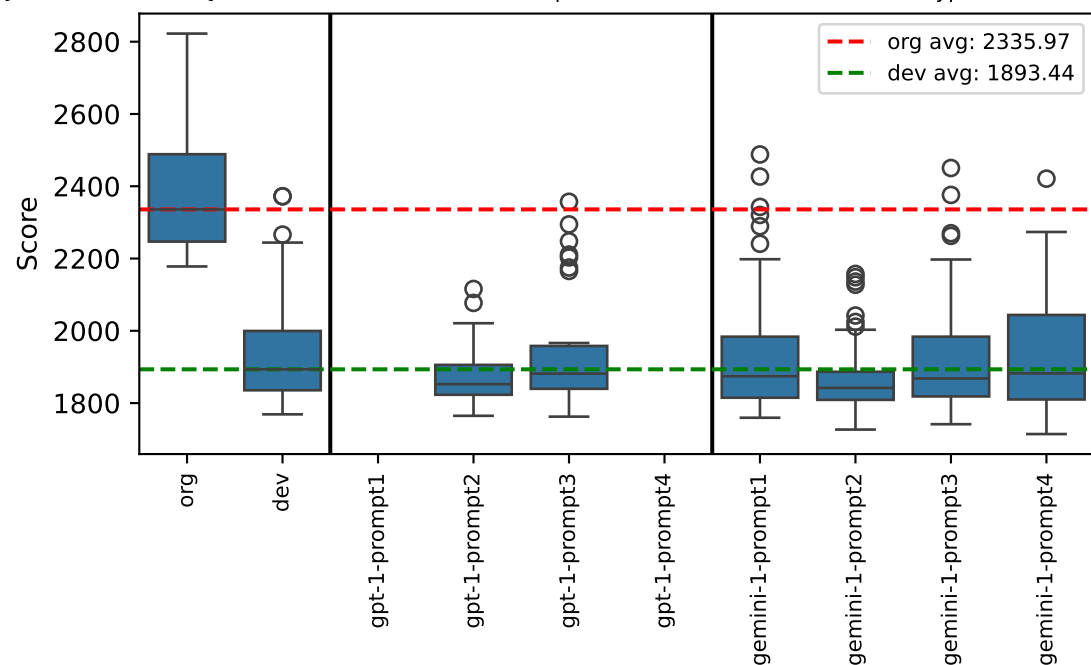
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'INTEGER'}



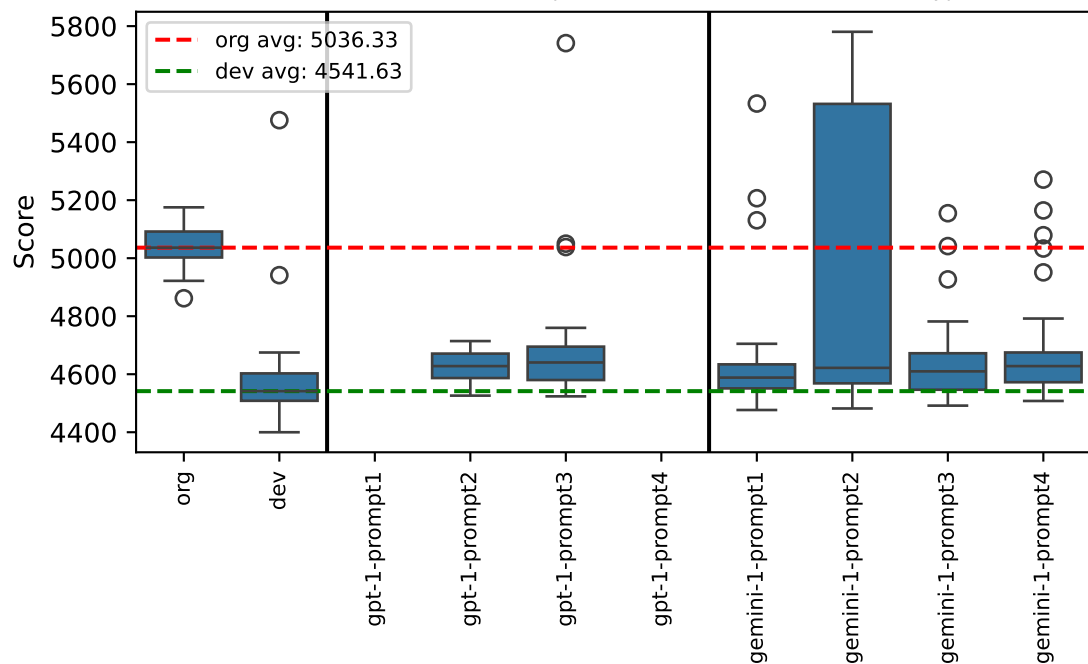
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'SMALLINT'}



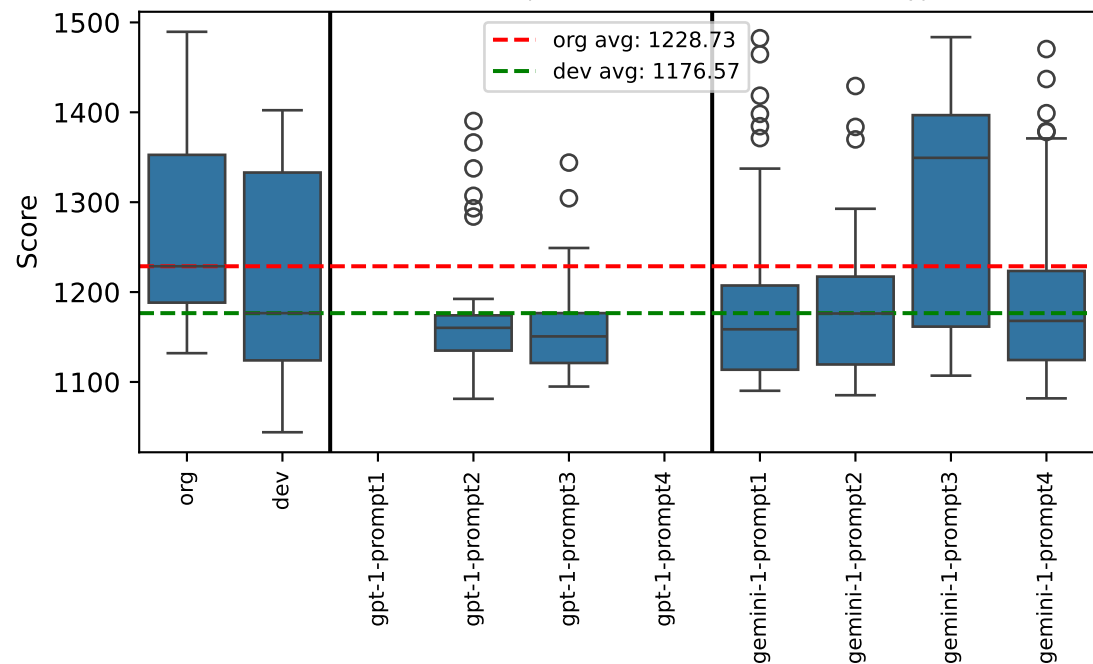
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'BOOLEAN'}



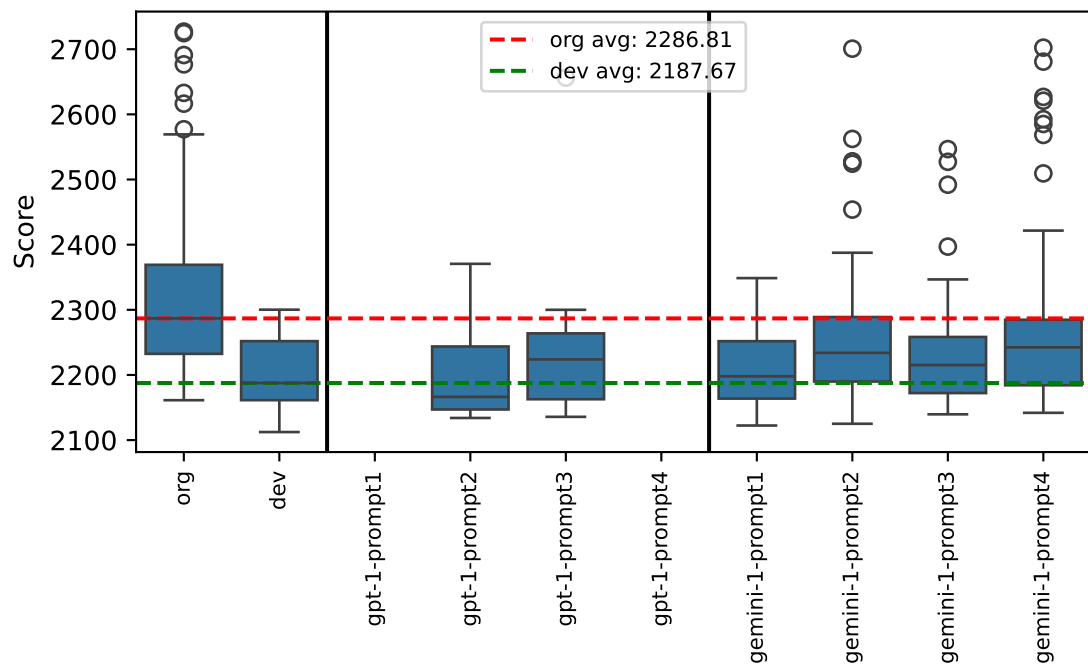
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'VARCHAR'}



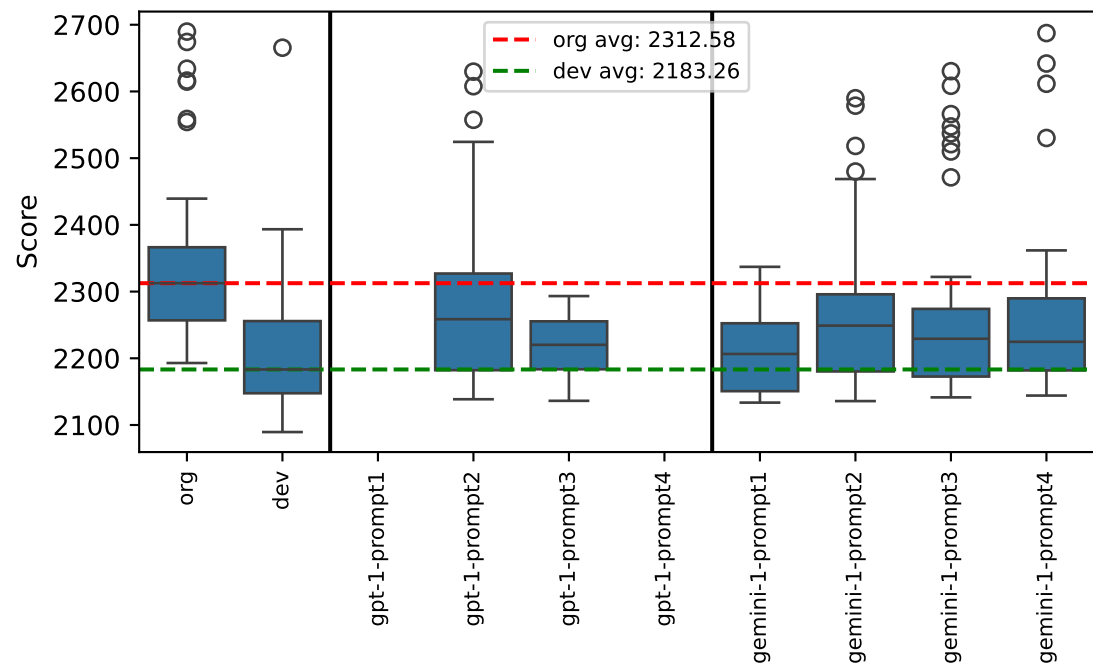
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(BIGINT)



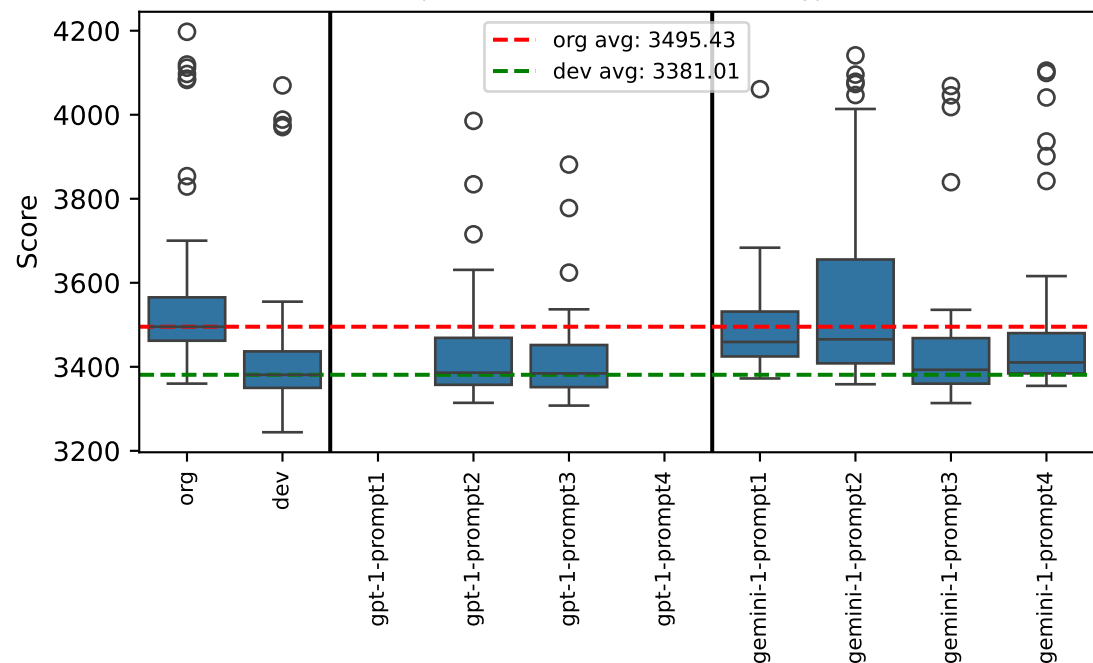
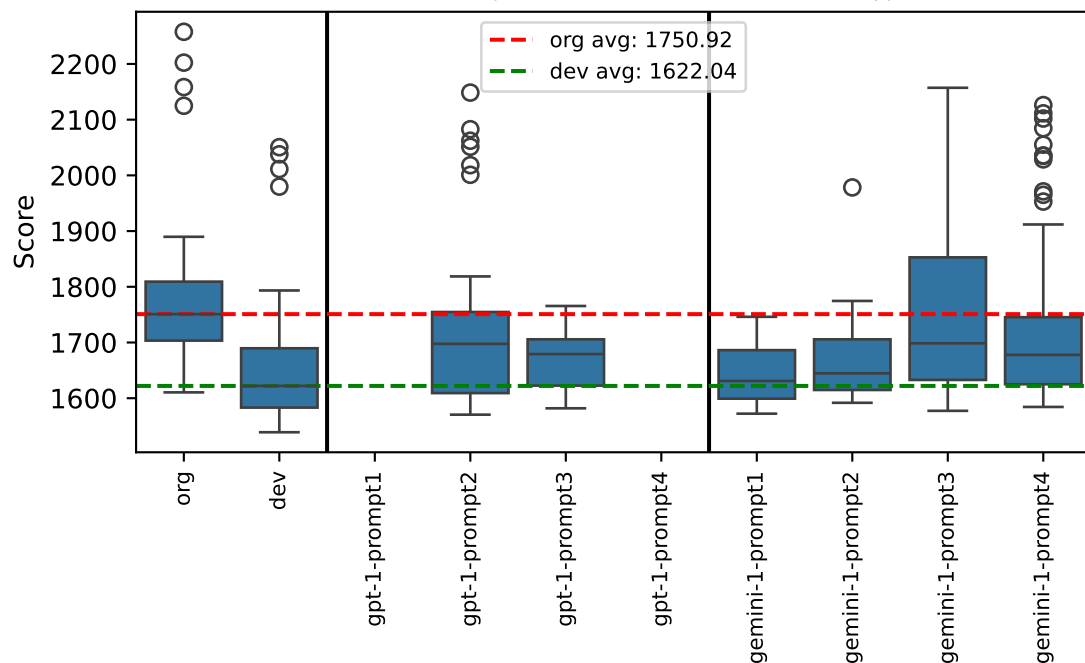
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(VARCHAR)



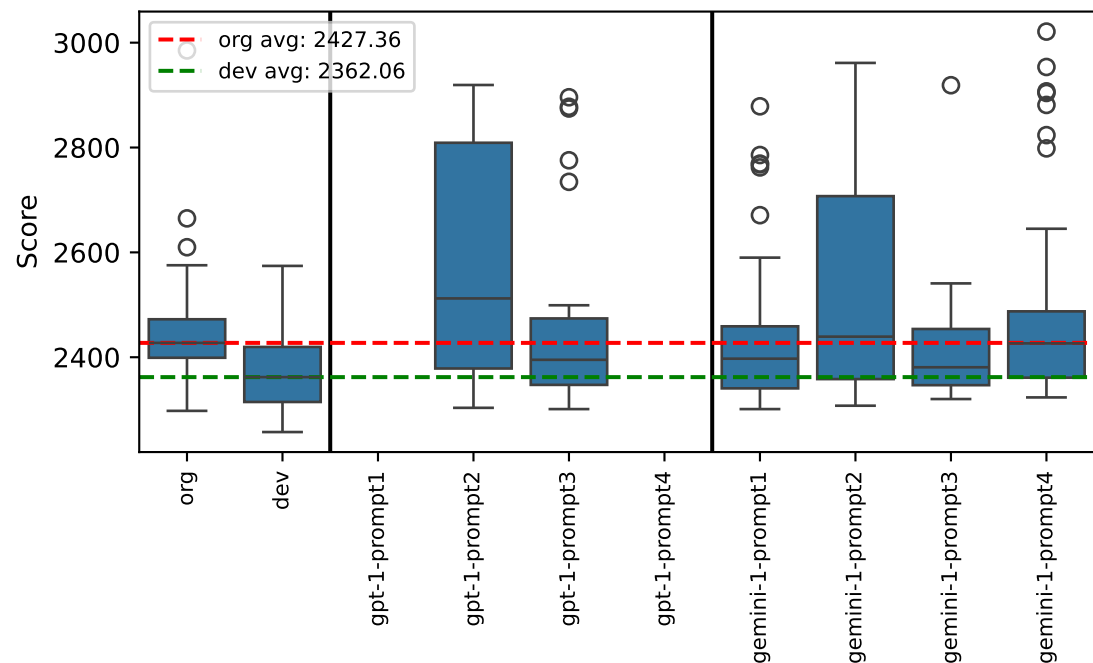
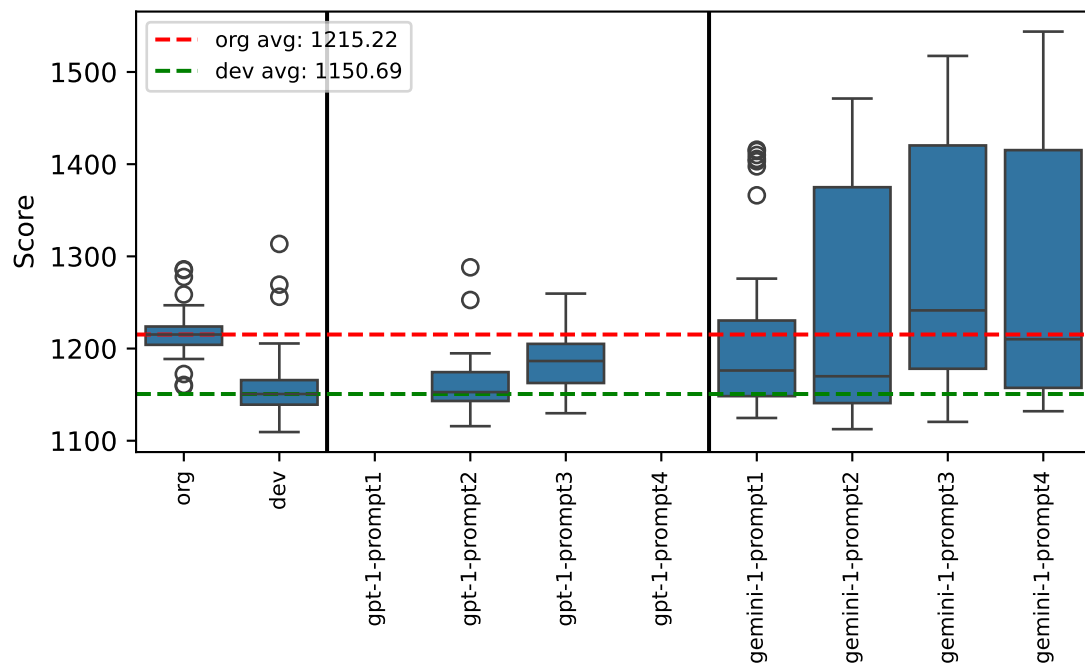
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(ARRAY(BIGINT



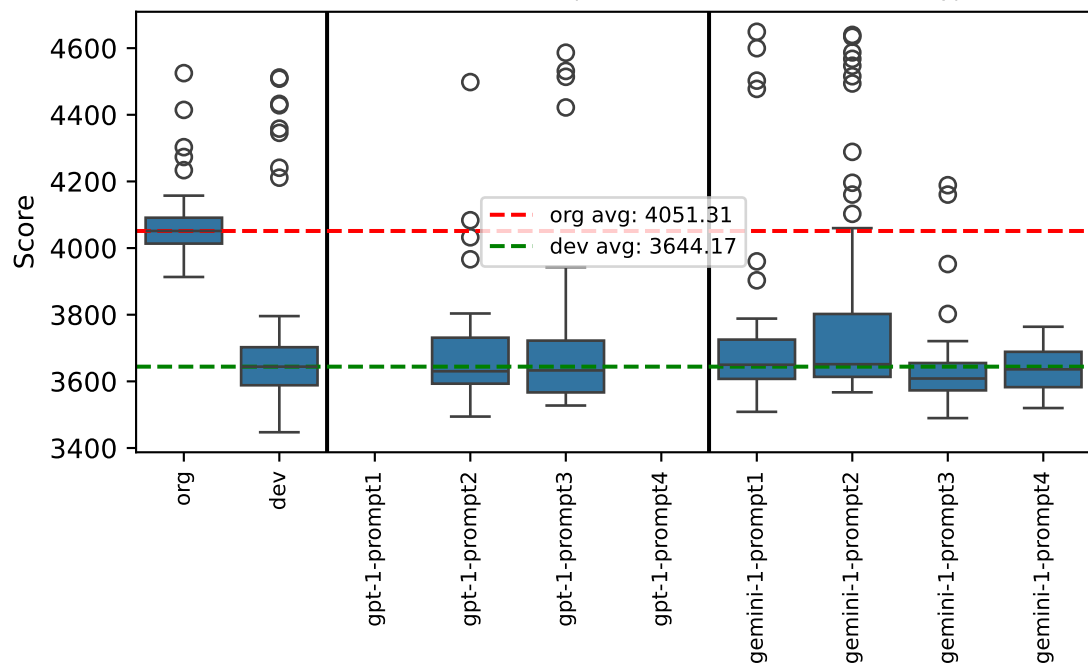
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'MAP(BIGINT, BIGINT)'} Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'MAP(BIGINT, MAP(BIGINT, BIGINT))'}



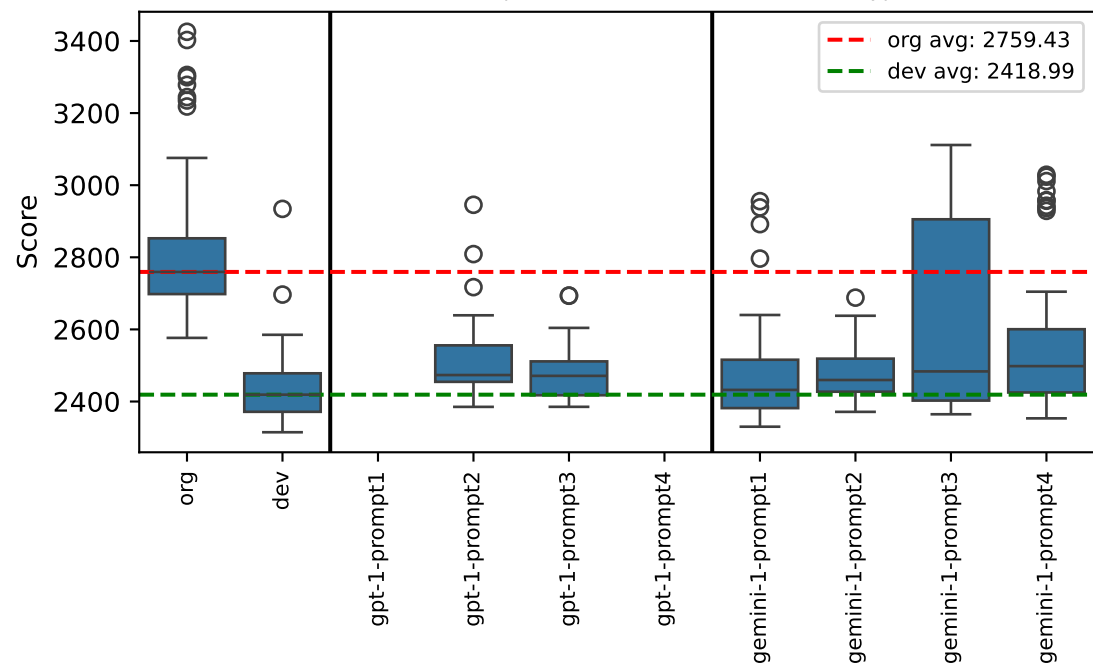
Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ROW(BIGINT, BIGINT)'} Params: {'channelCount': '1', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'}



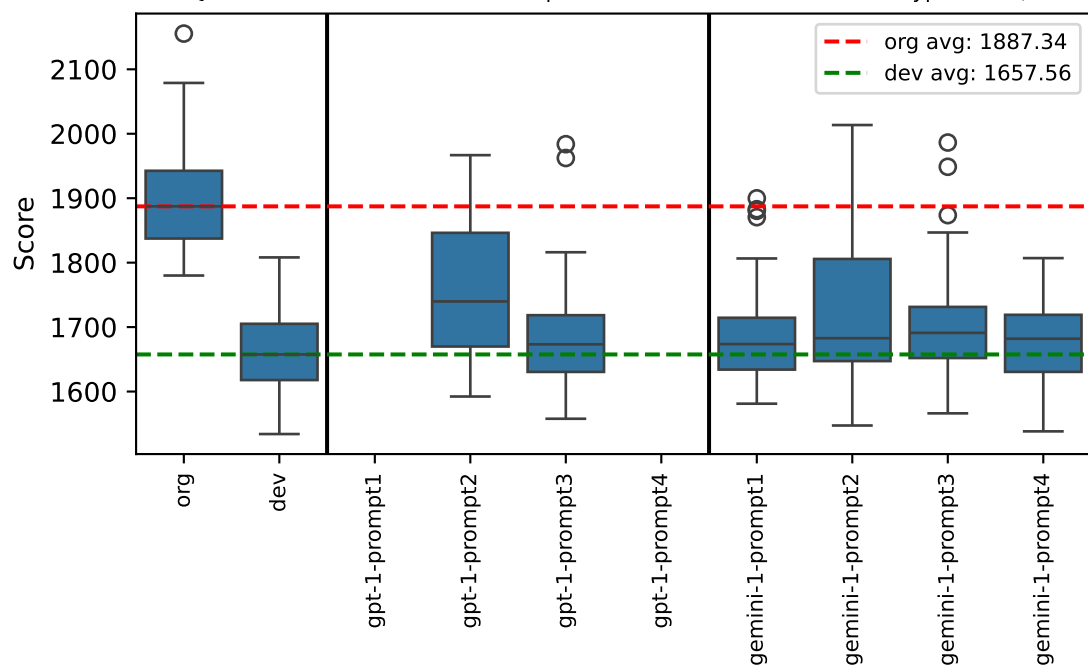
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'BIGINT'}



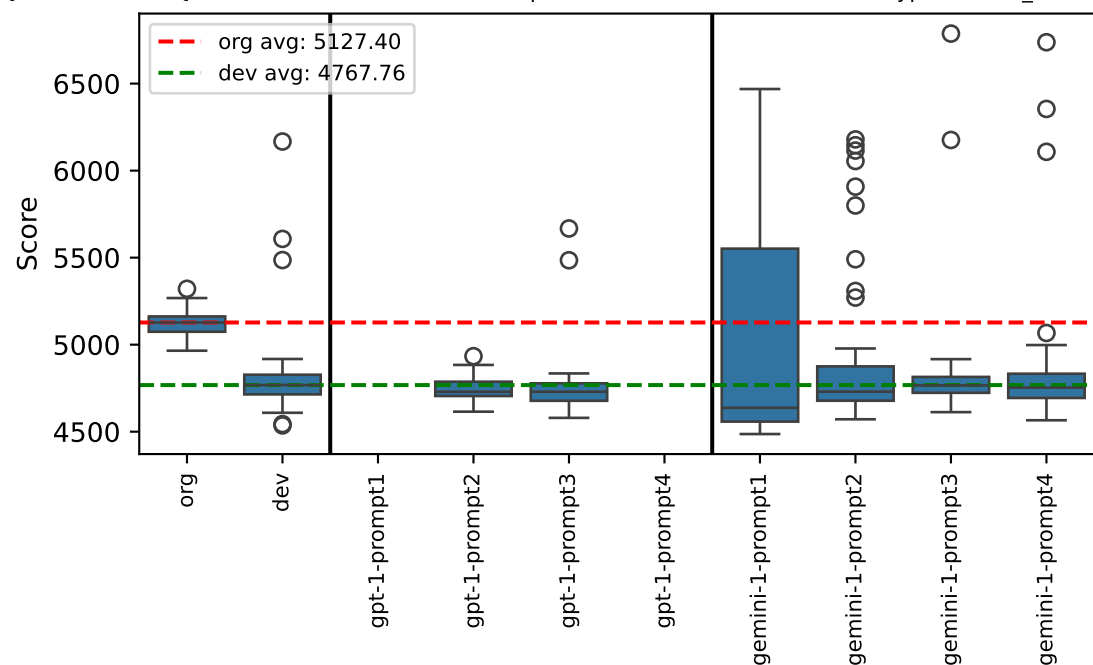
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'DICTIONARY(BIGINT, BIGINT)'}

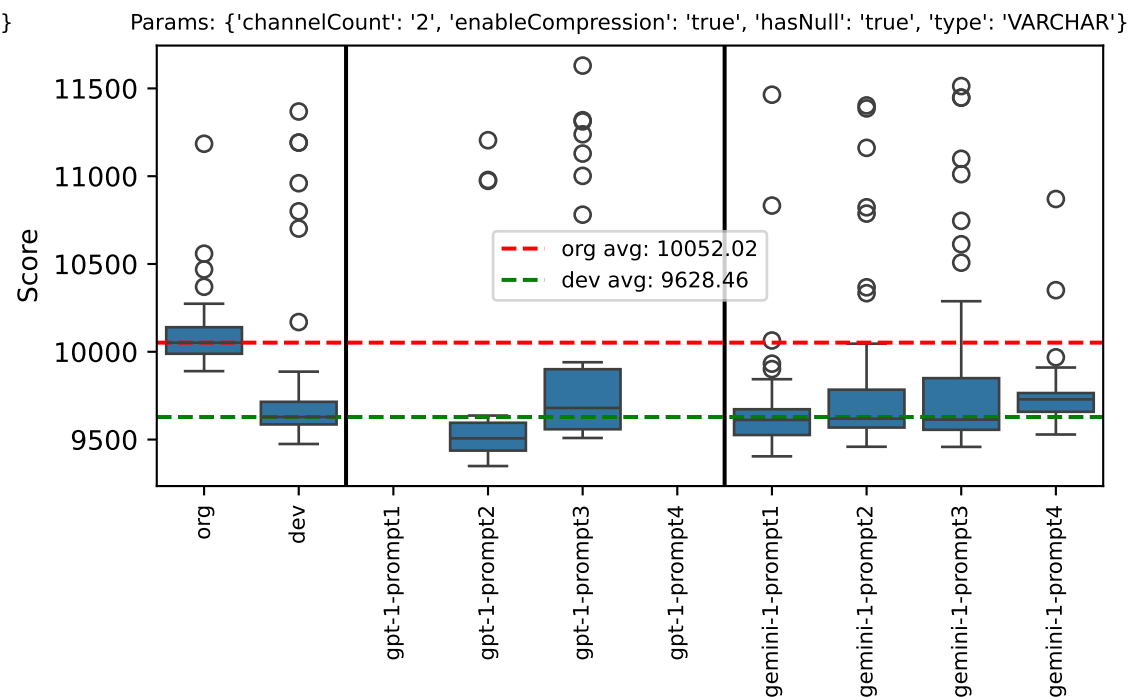
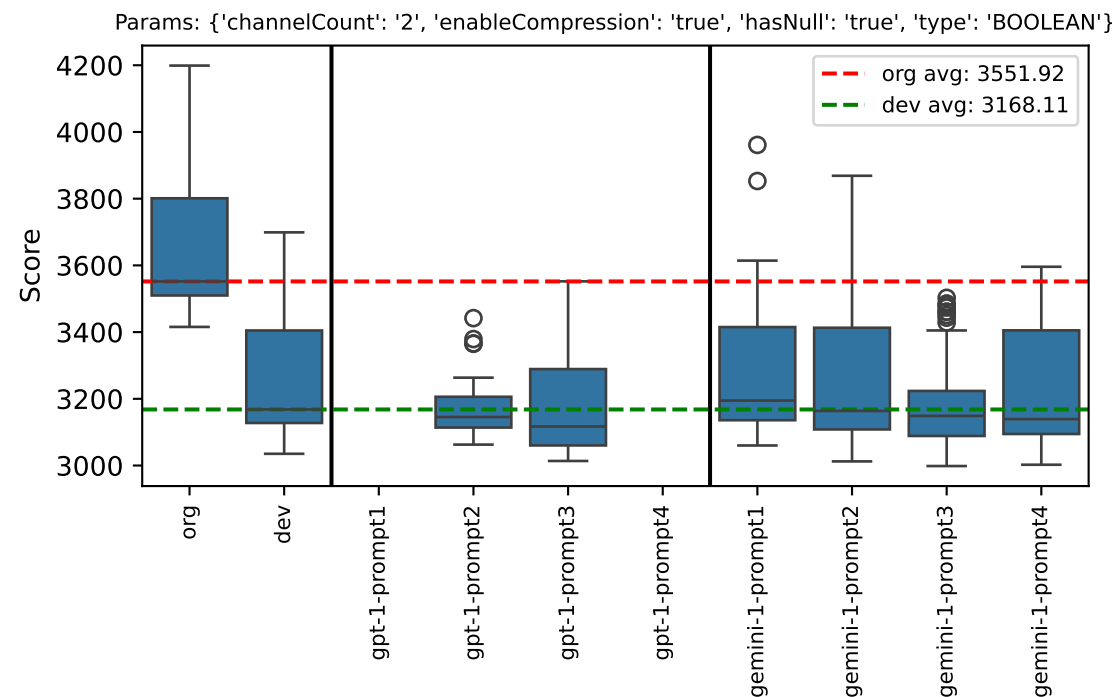
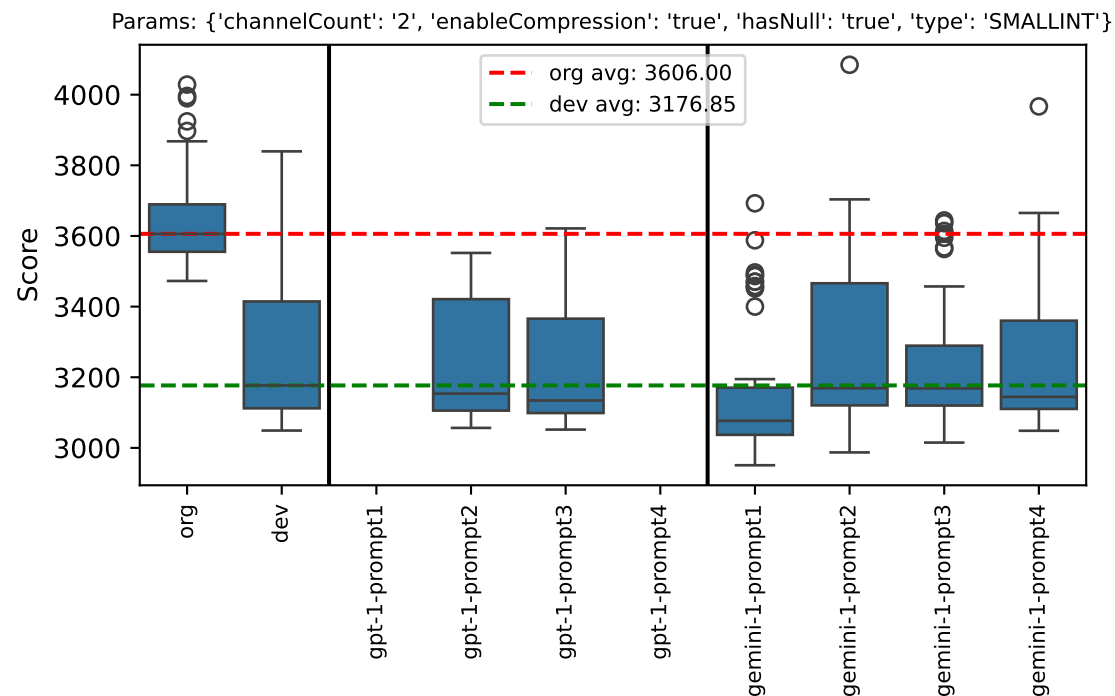
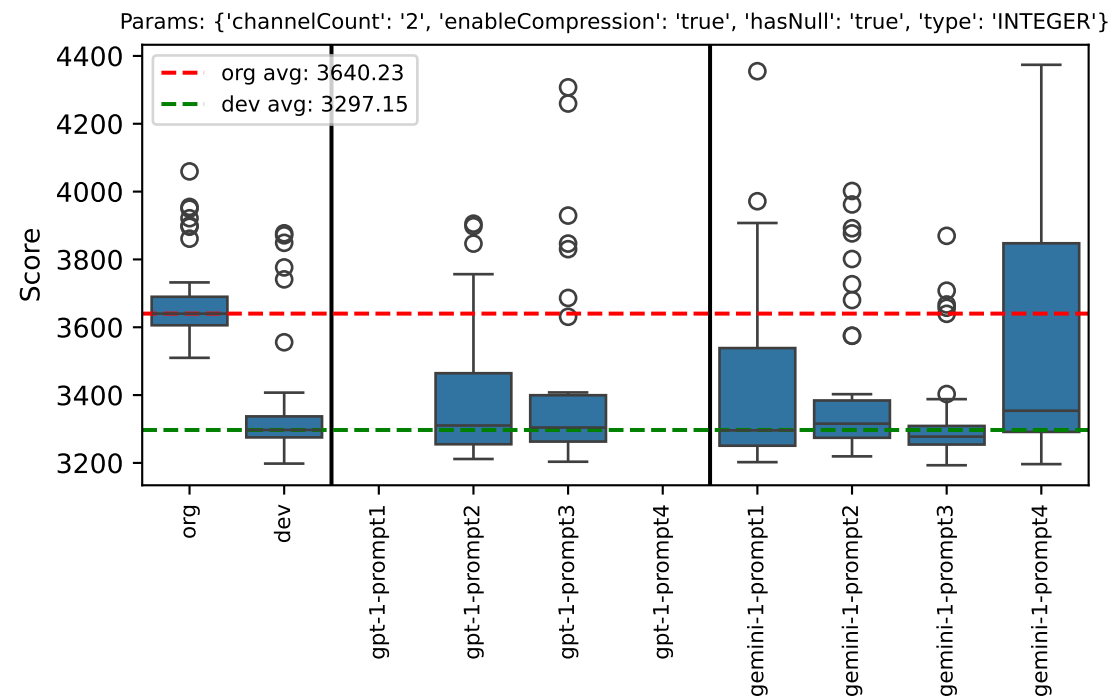


Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'RLE(BIGINT)'}

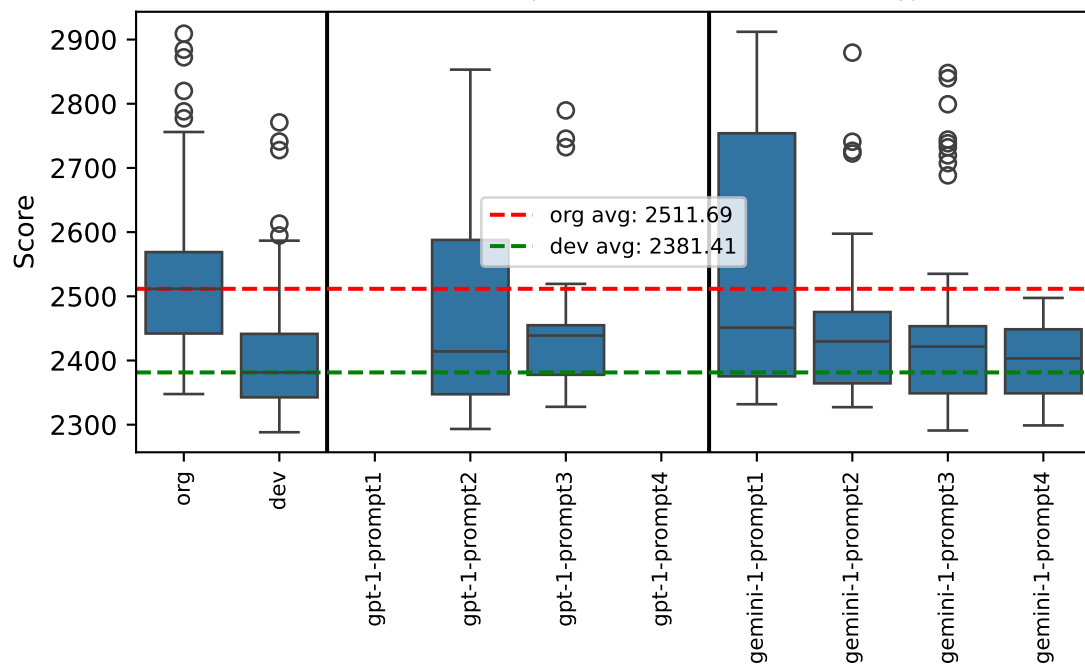


Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'LONG_DECIMAL'}

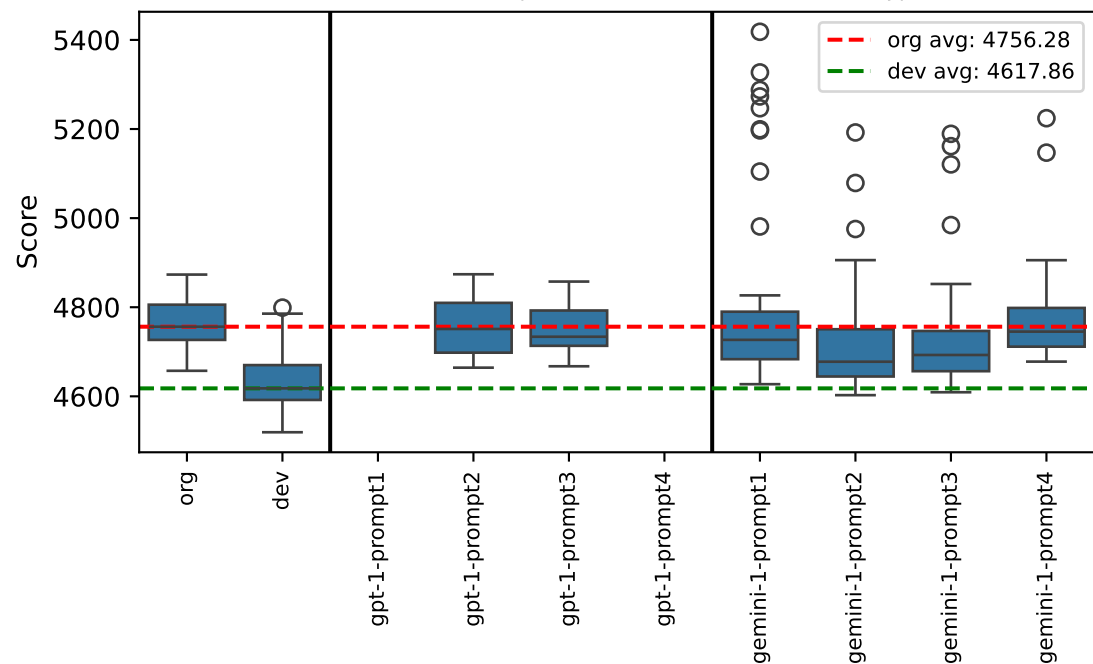




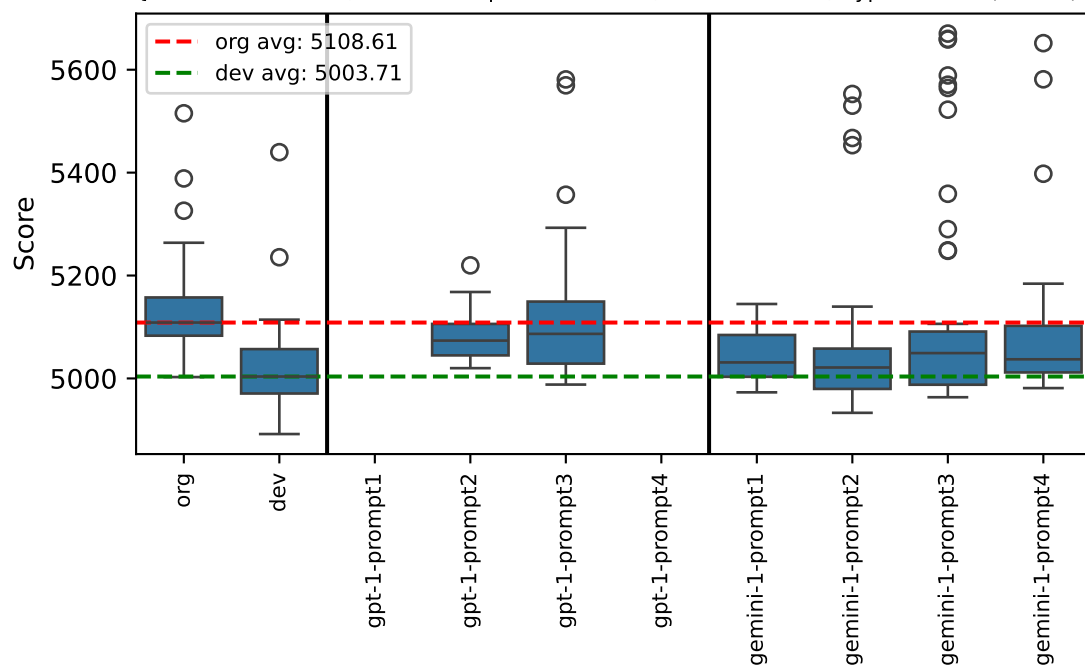
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(BIGINT)'}



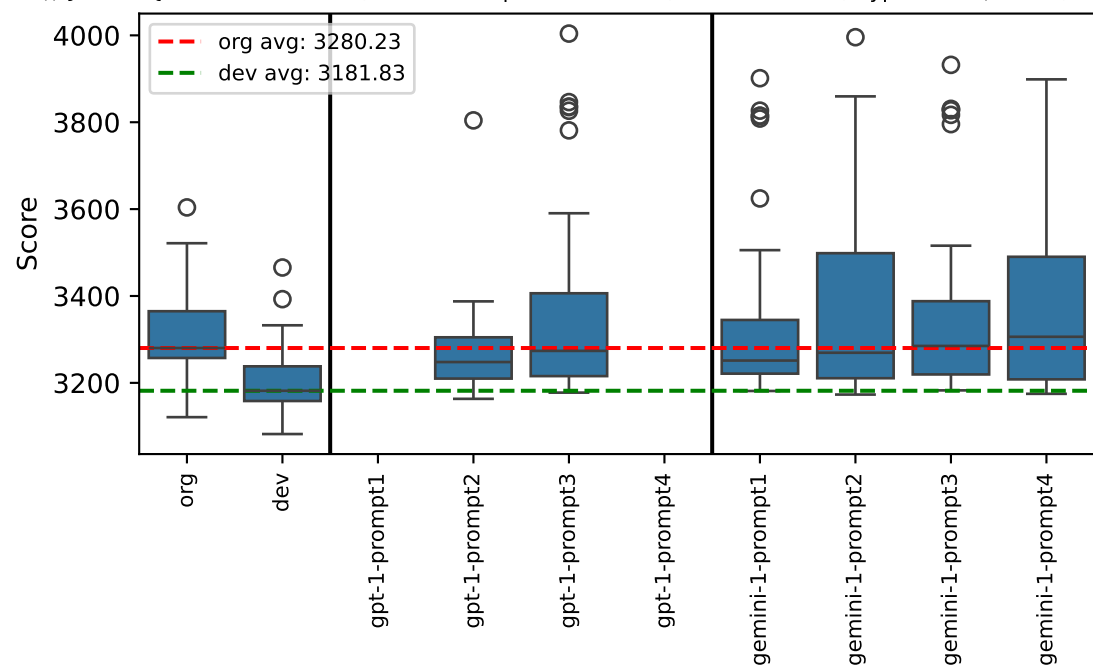
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(VARCHAR(10))'}



Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ARRAY(ARRAY(BIGINT))'}

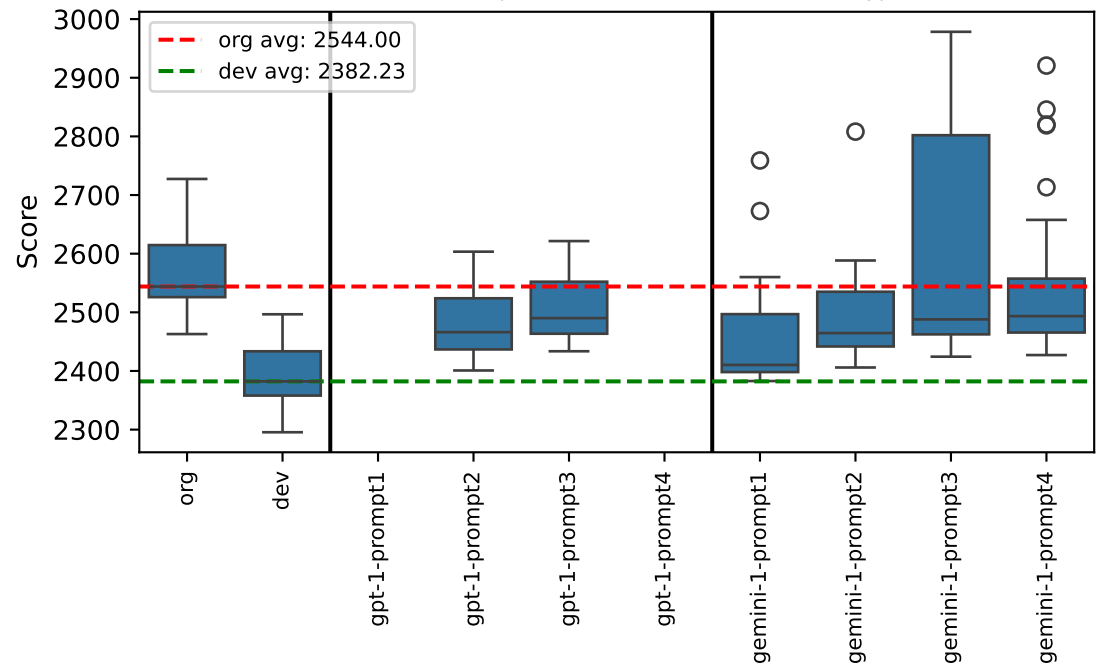
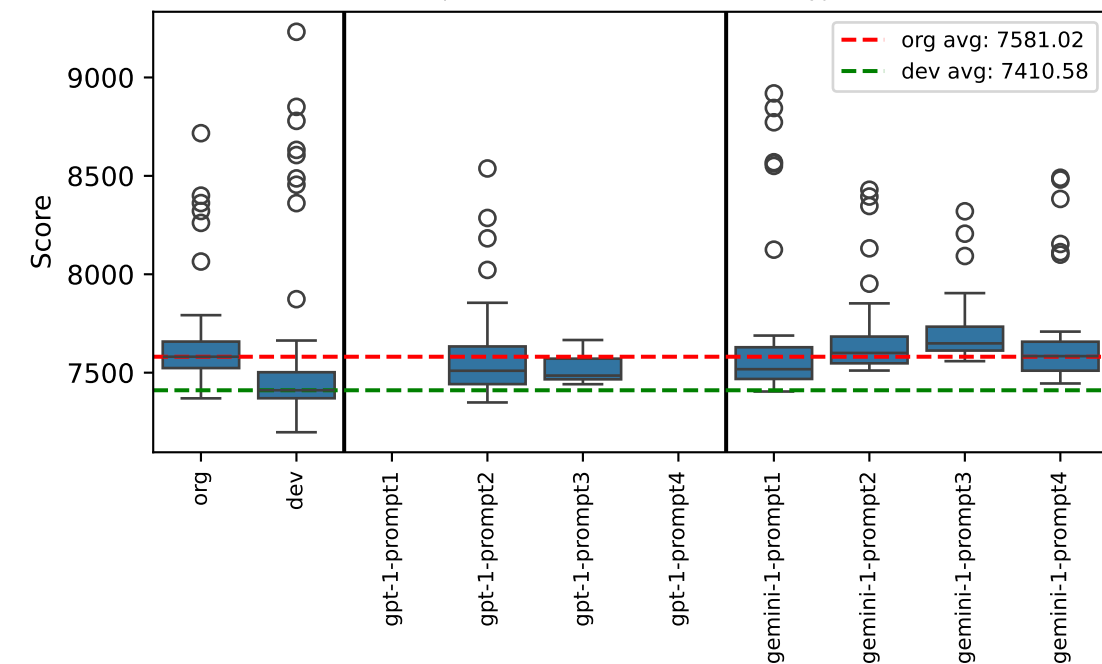


Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'MAP(BIGINT, BIGINT)'}

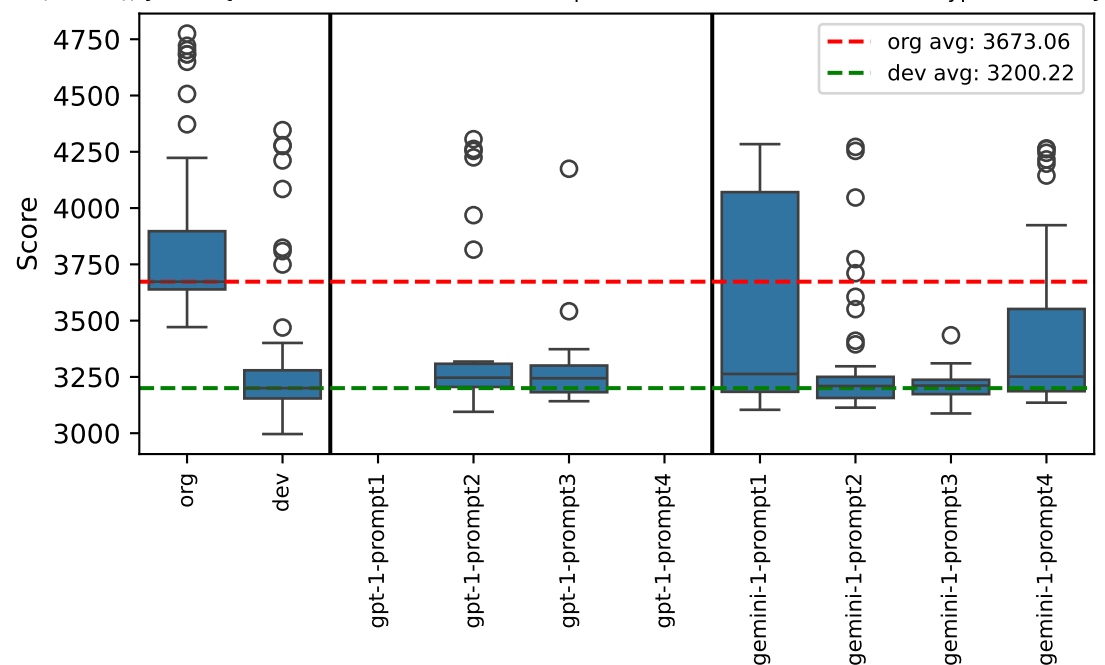
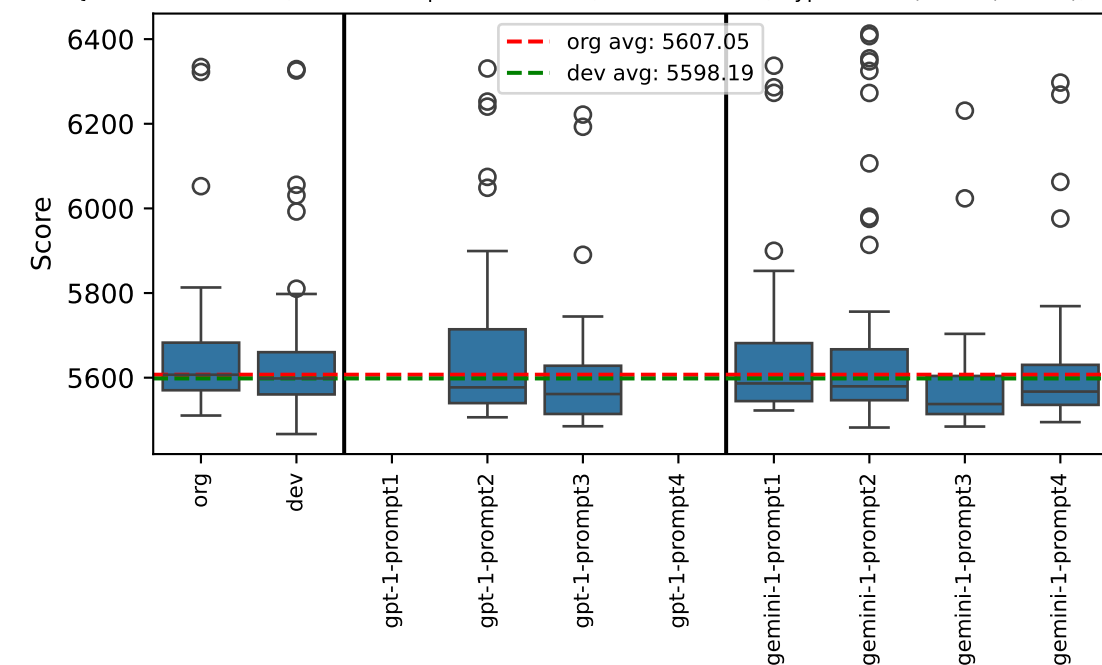


Benchmark: com.facebook.presto.operator.repartition.BenchmarkPartitionedOutputOperator.optimizedAddPa

Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'MAP(BIGINT, MAP(BIGINT, BIGINT))'} Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ROW(BIGINT, BIGINT)'} Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'} Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'BIGINT'}

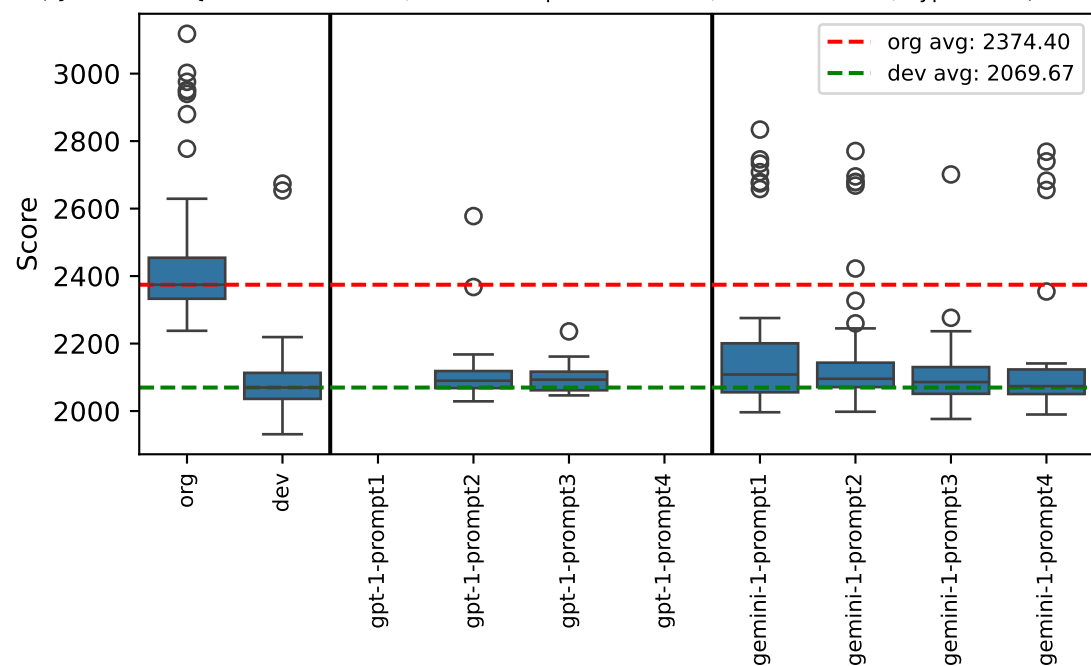
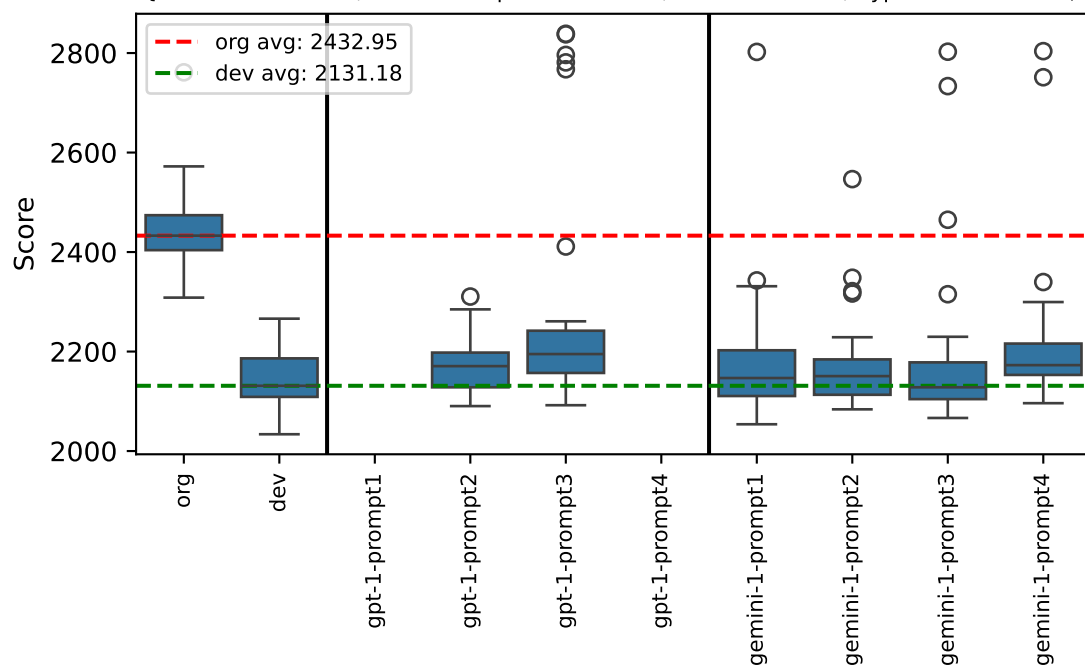


Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'true', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'} Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'BIGINT'}



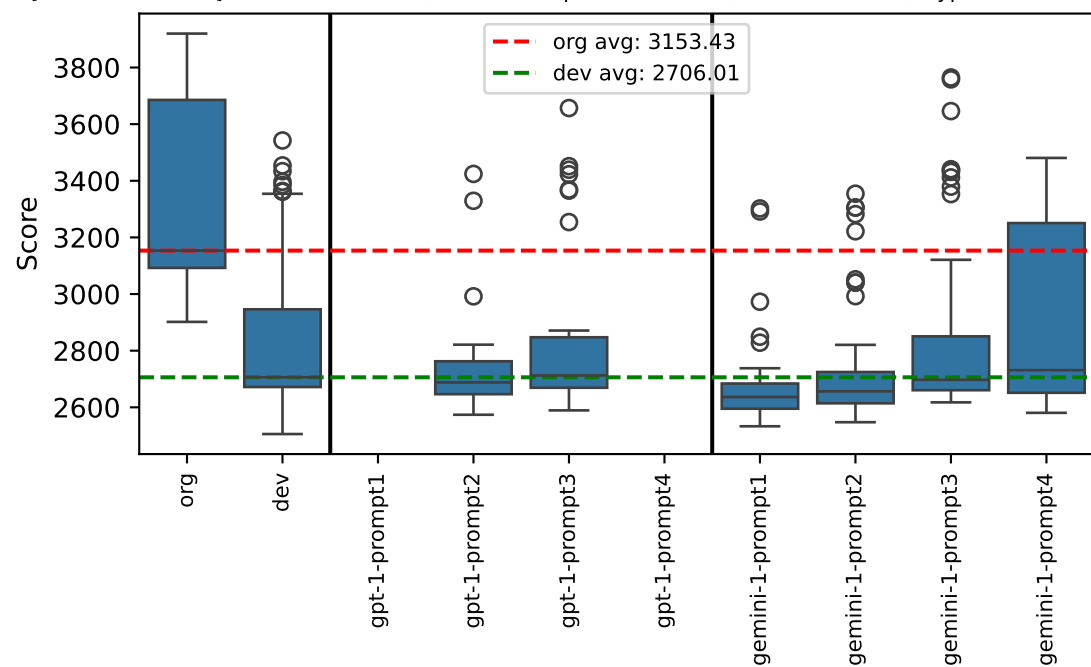
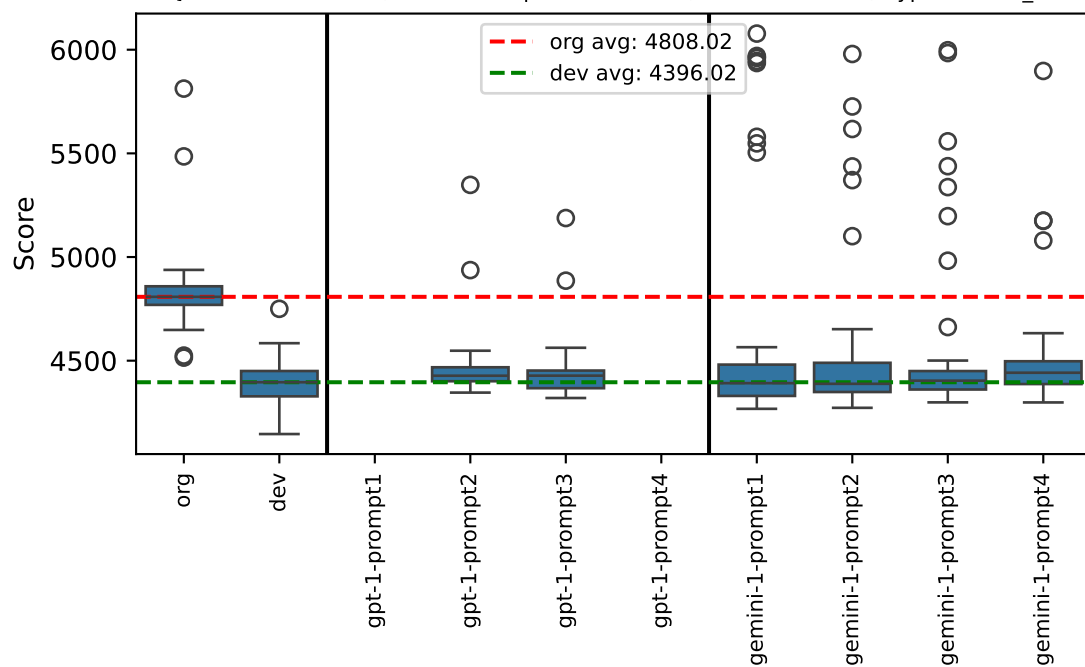
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'DICTIONARY(BIGINT)'}

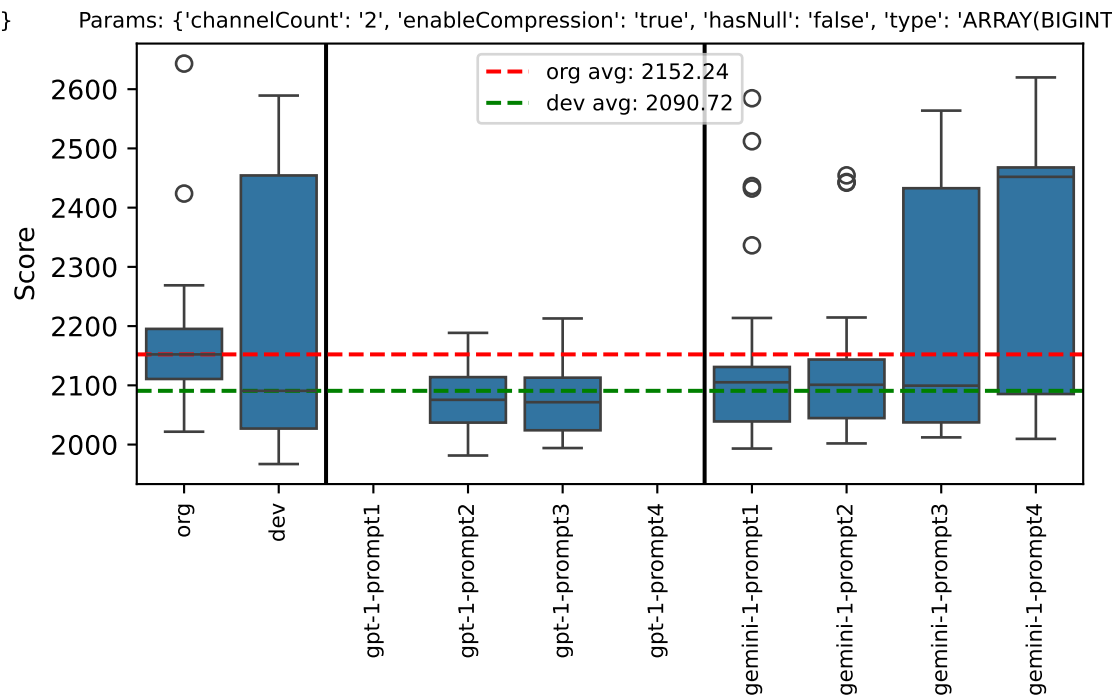
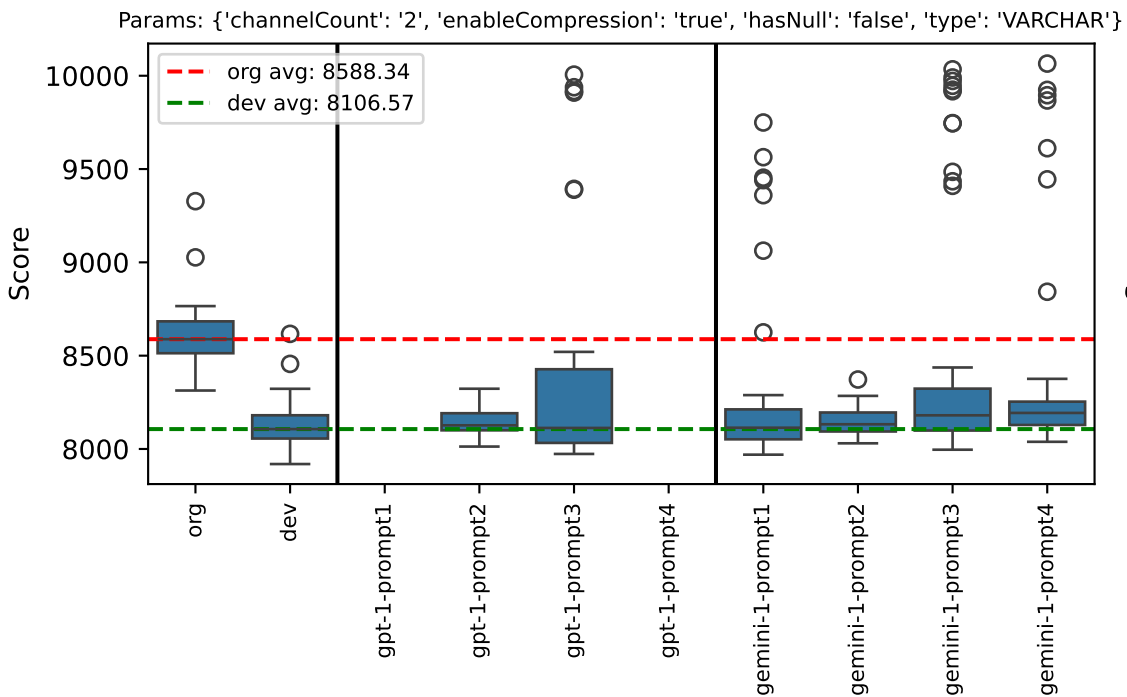
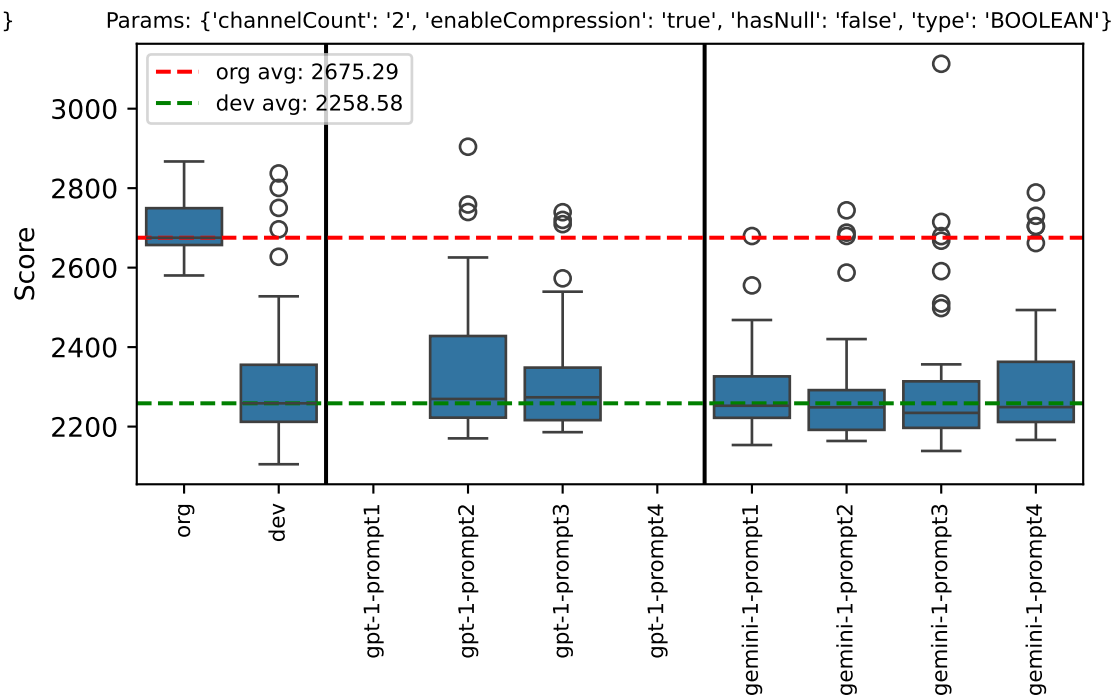
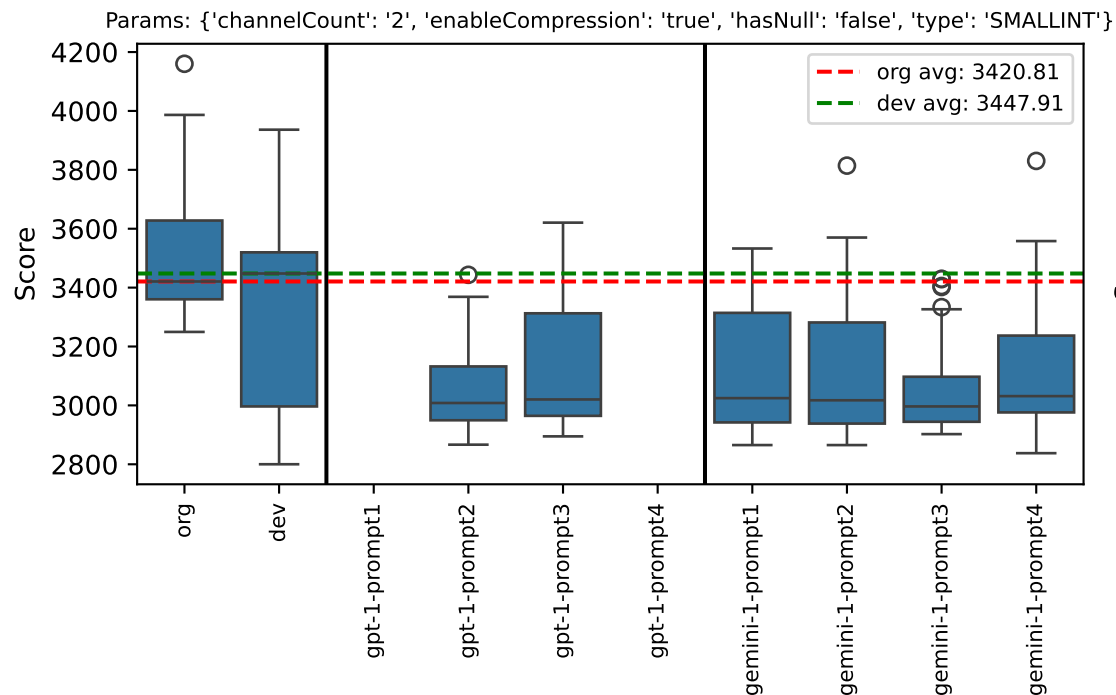
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'RLE(BIGINT)'}

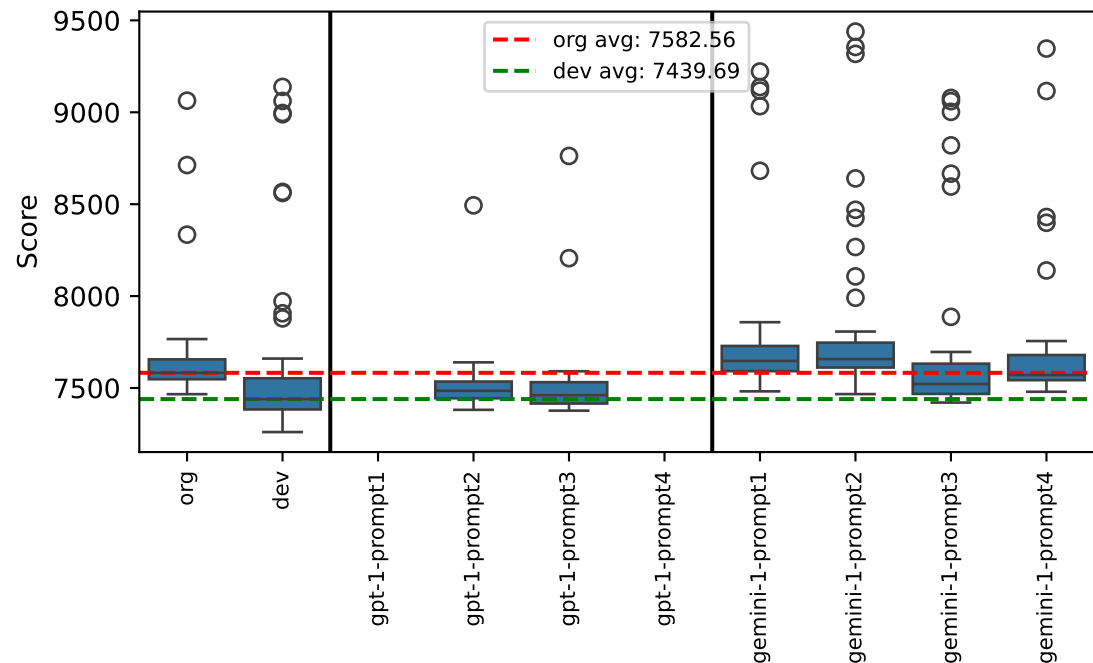
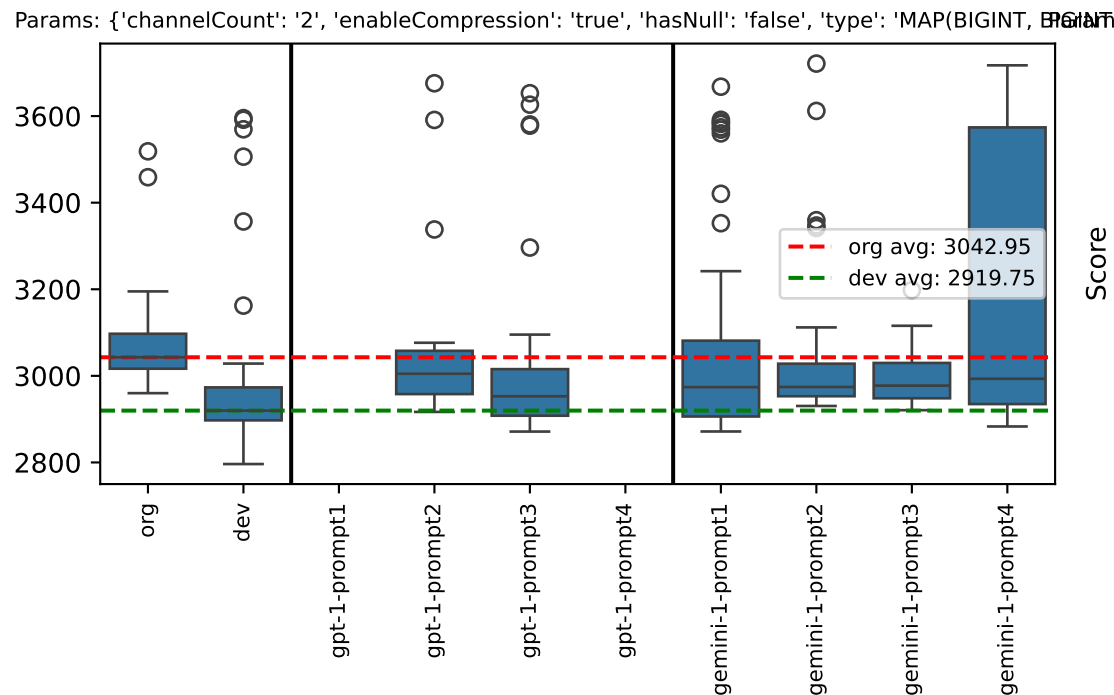
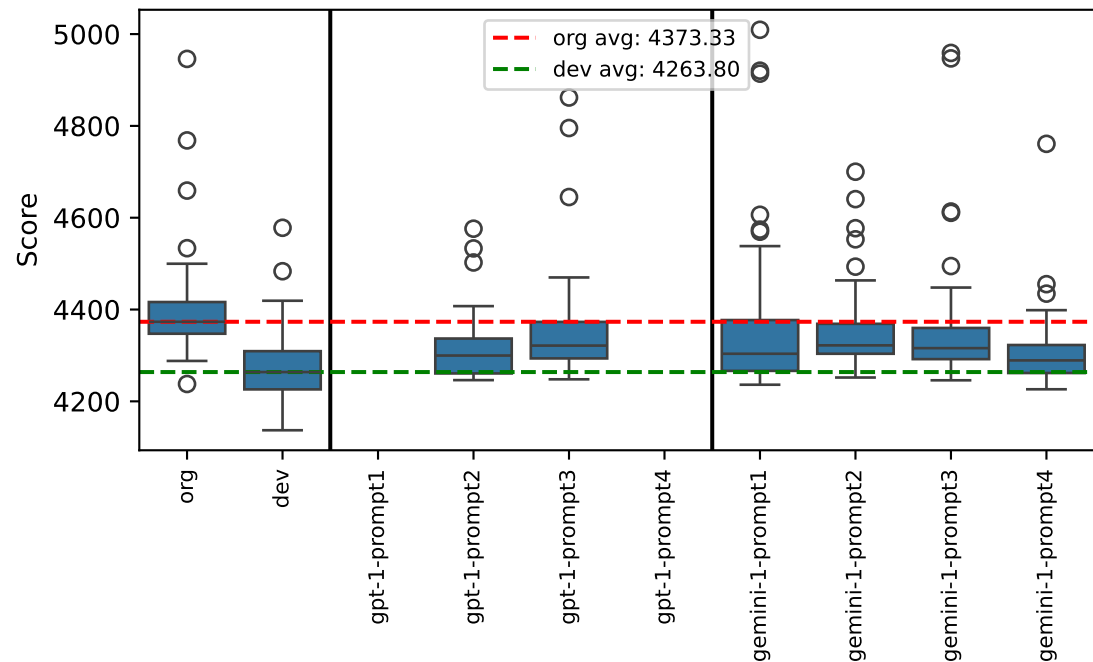
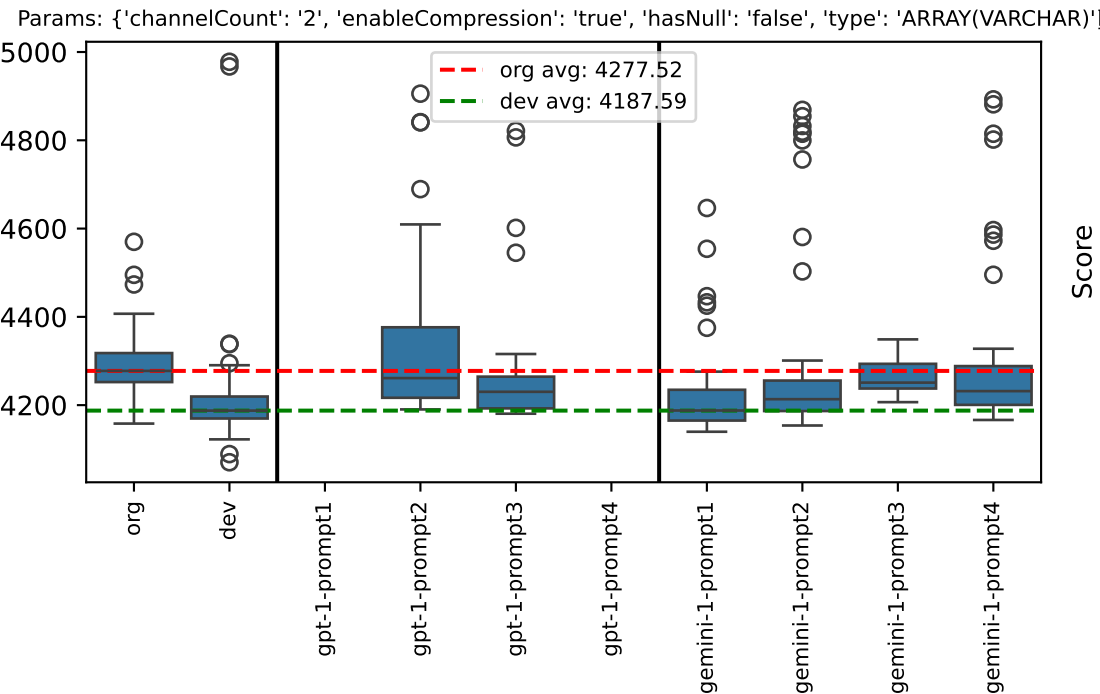


Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'LONG_DECIMAL'}

Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'INTEGER'}

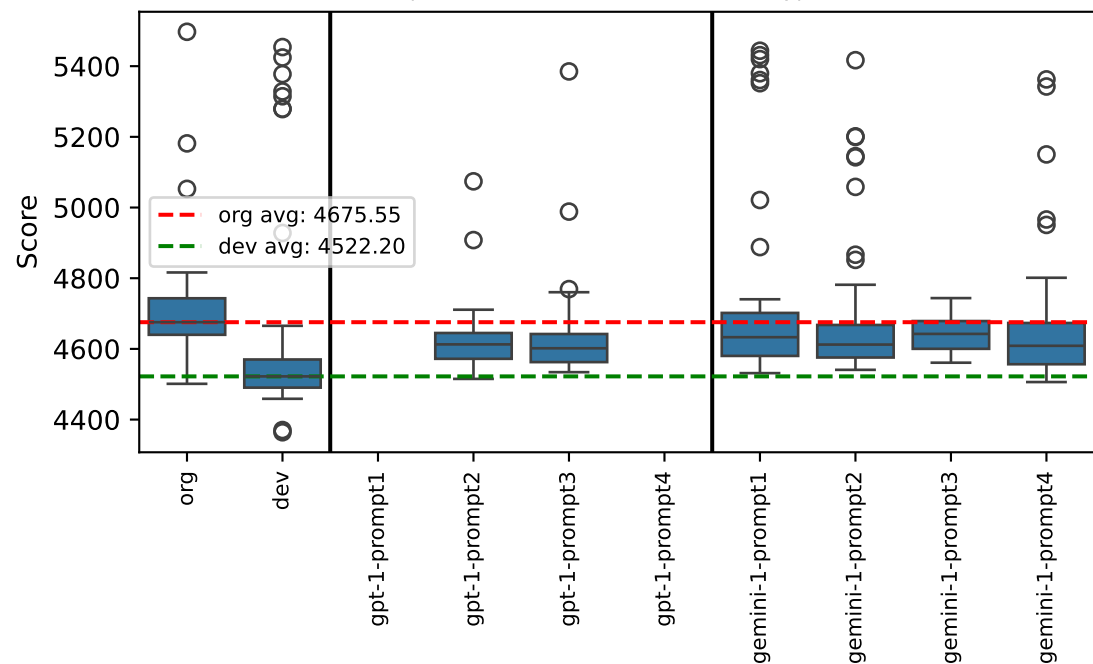
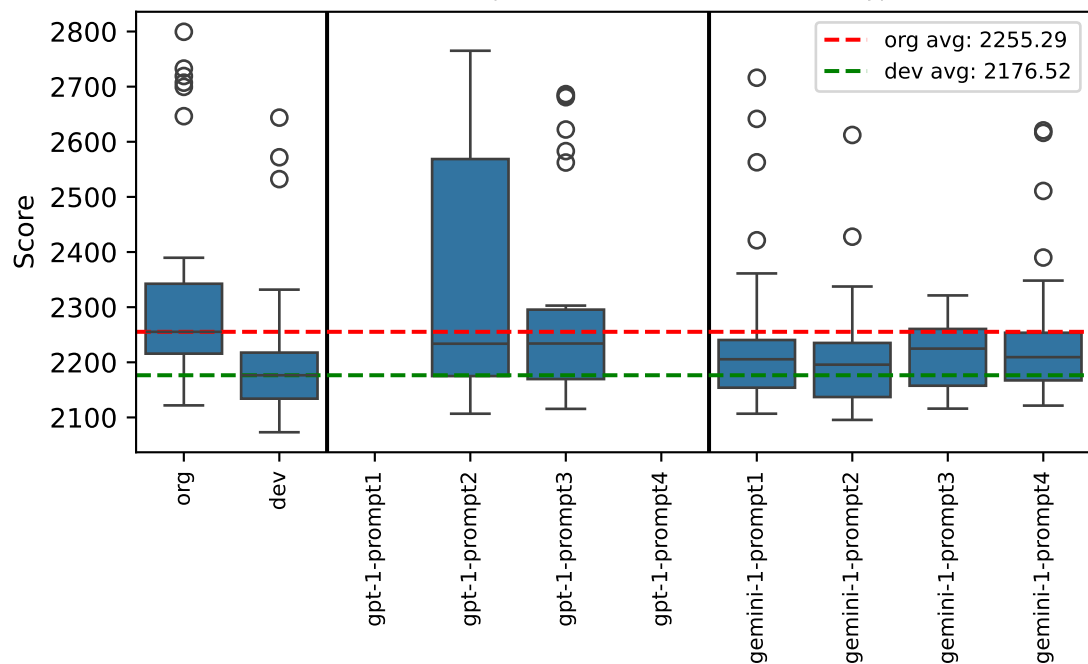




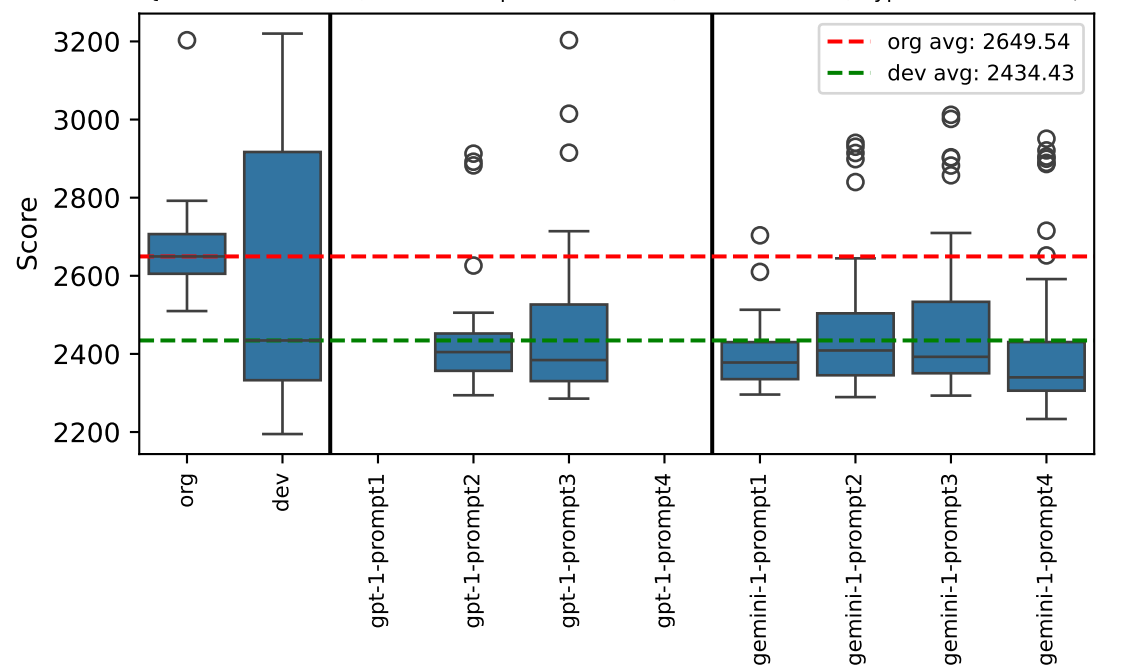
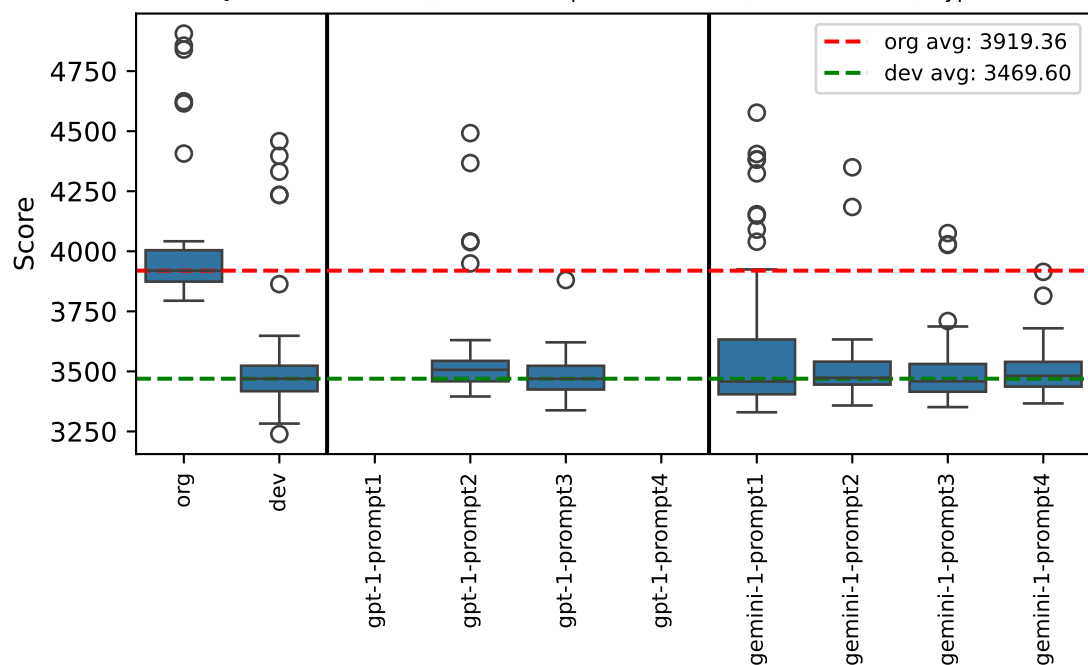


enchmark: com.facebook.presto.operator.repartition.BenchmarkPartitionedOutputOperator.optimizedAddPa

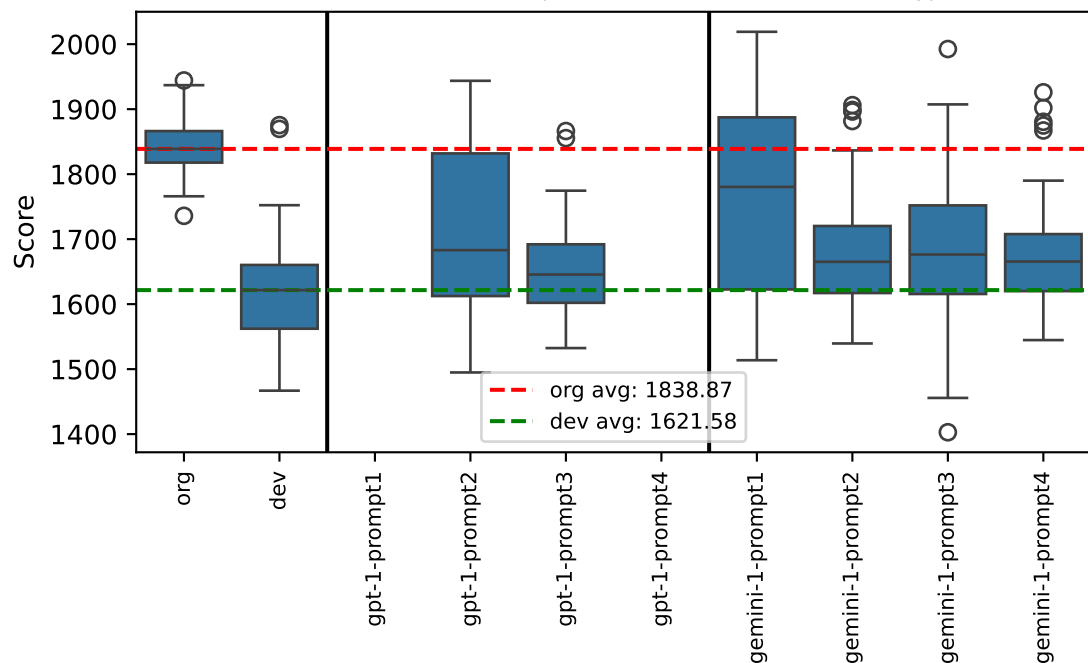
Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'ROW(BIGINT, BIGINT)'} Params: {'channelCount': '2', 'enableCompression': 'true', 'hasNull': 'false', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'}



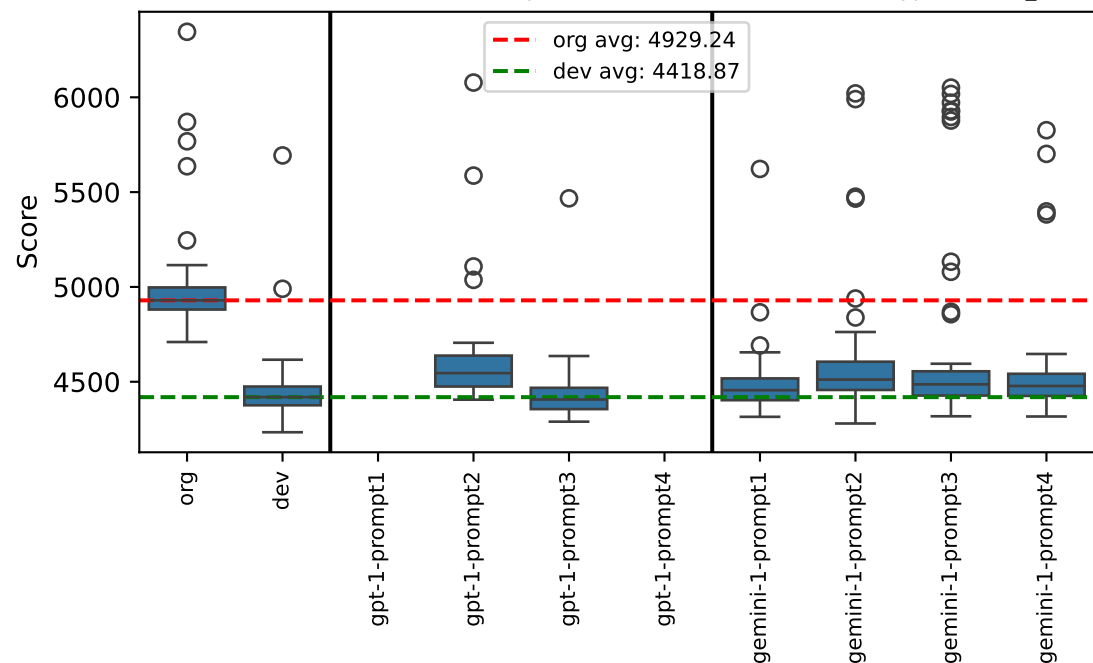
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'BIGINT'}	Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'DICTIONARY(BIGINT)'}
--------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------



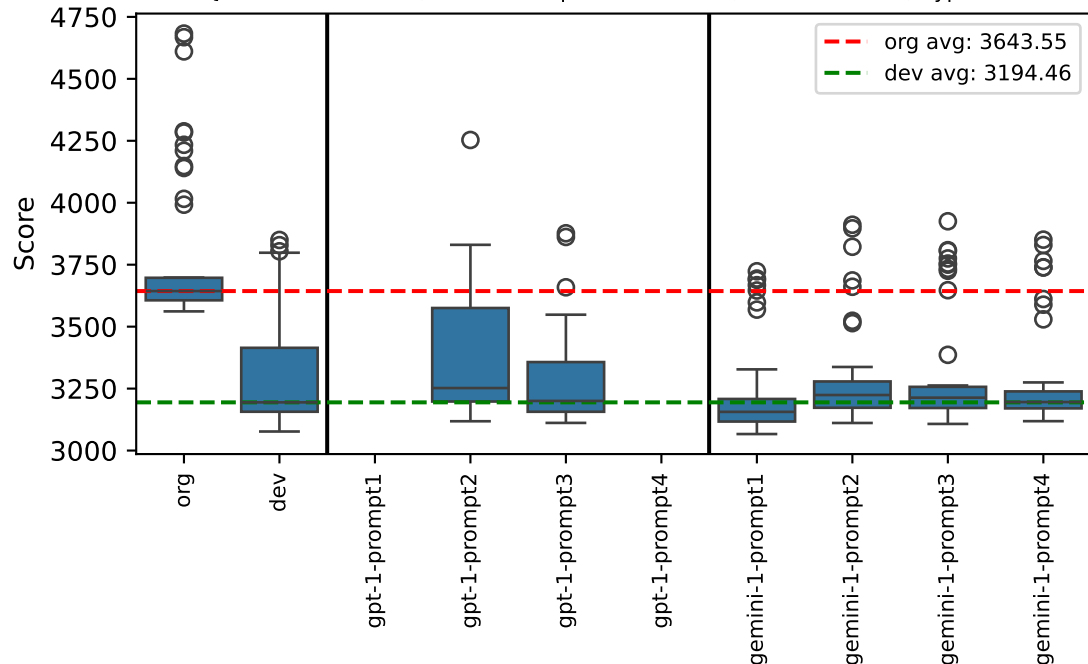
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'RLE(BIGINT)'}
org avg: 1838.87
dev avg: 1621.58



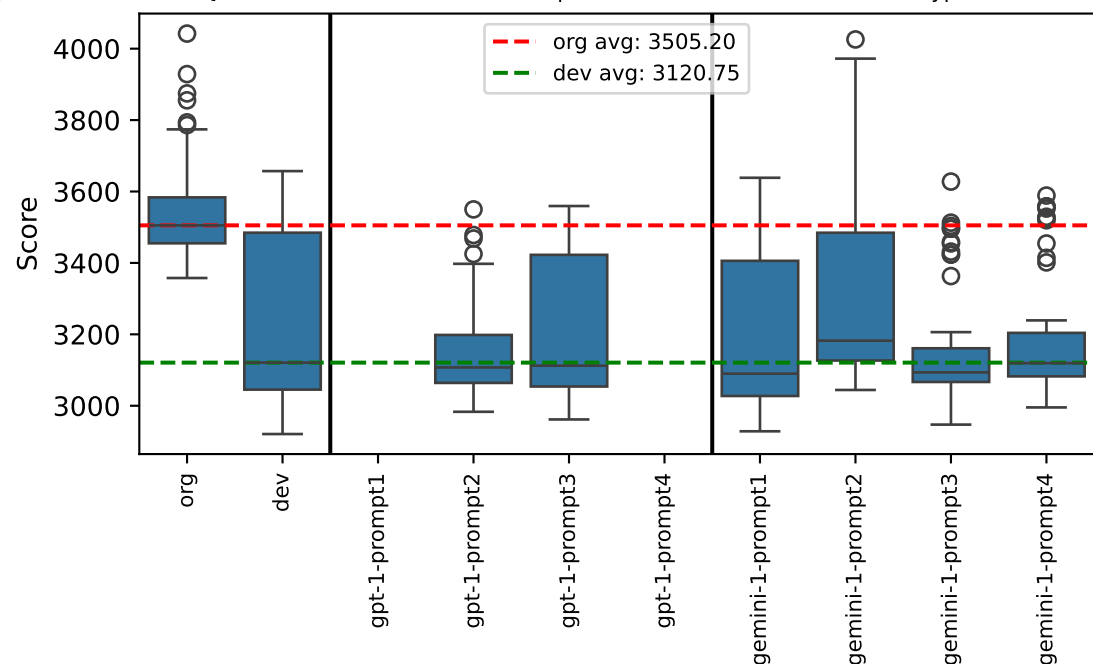
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'LONG_DECIMAL'}
org avg: 4929.24
dev avg: 4418.87



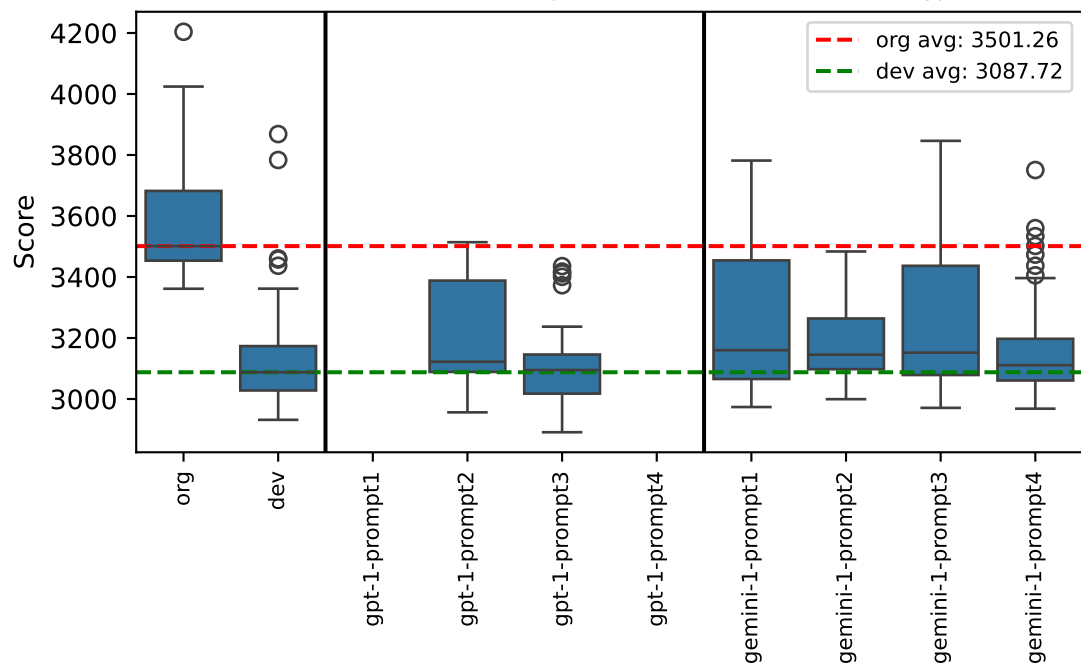
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'INTEGER'}
org avg: 3643.55
dev avg: 3194.46



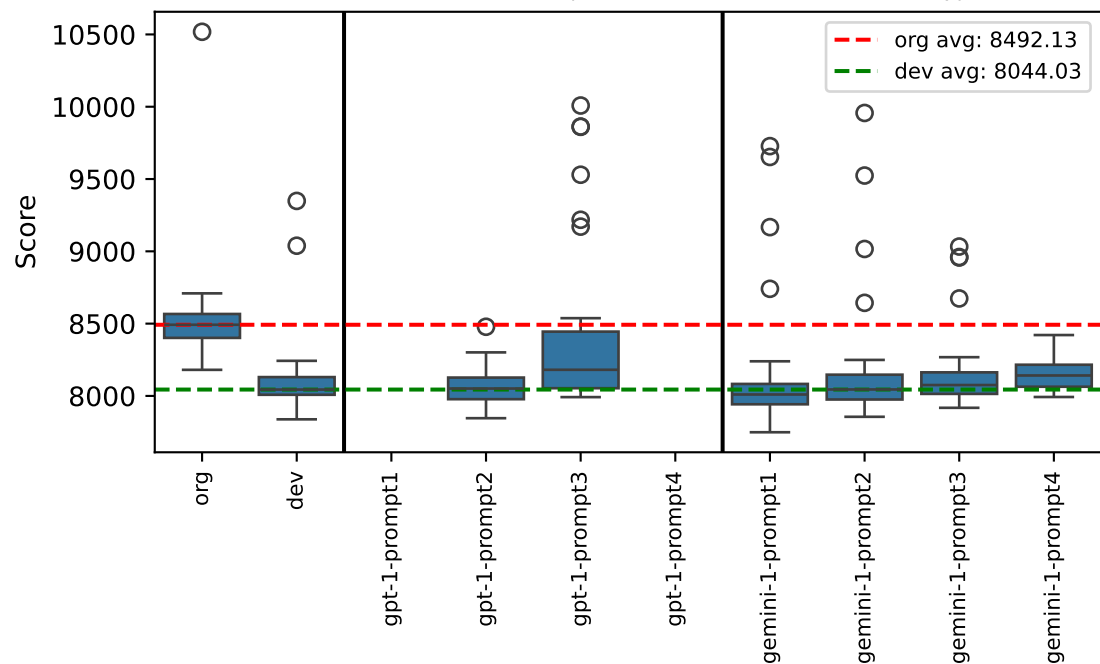
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'SMALLINT'}
org avg: 3505.20
dev avg: 3120.75



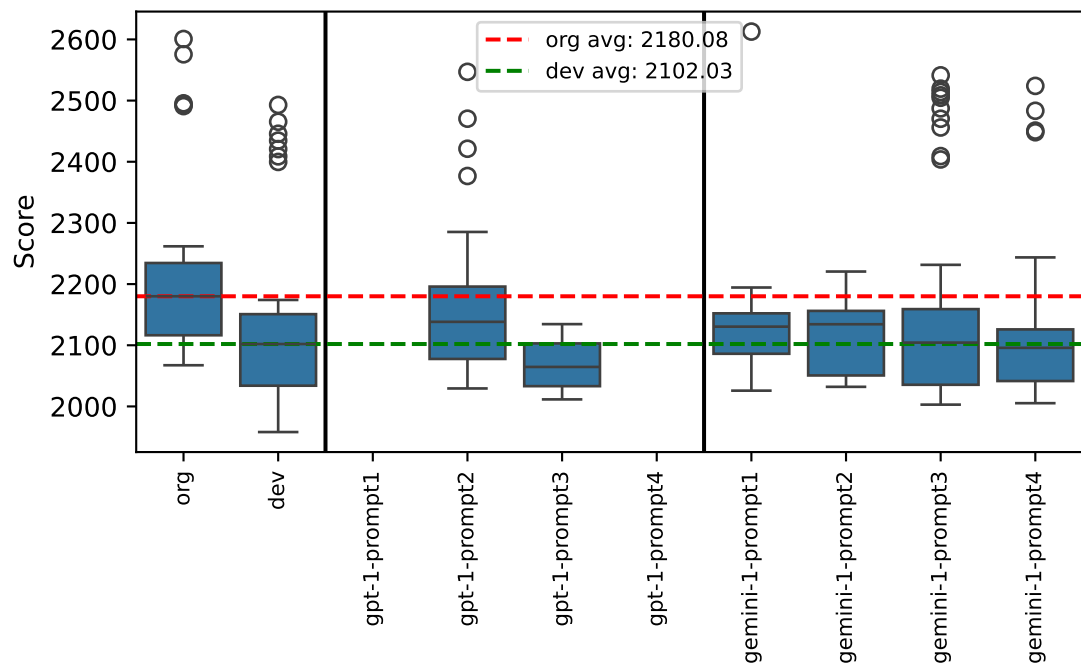
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'BOOLEAN'}



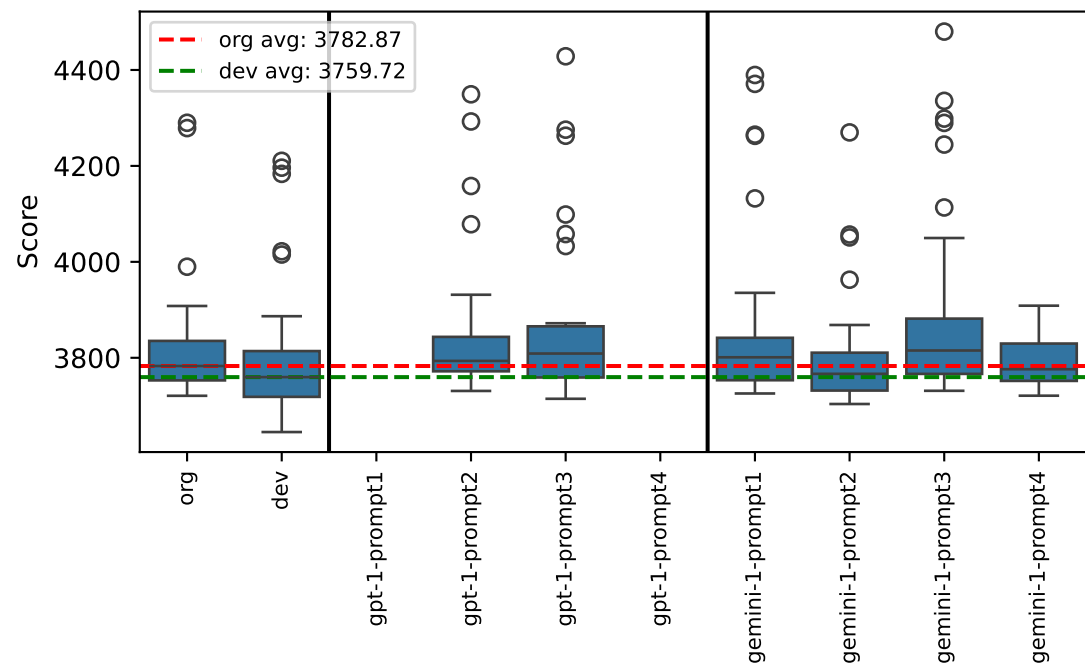
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'VARCHAR'}



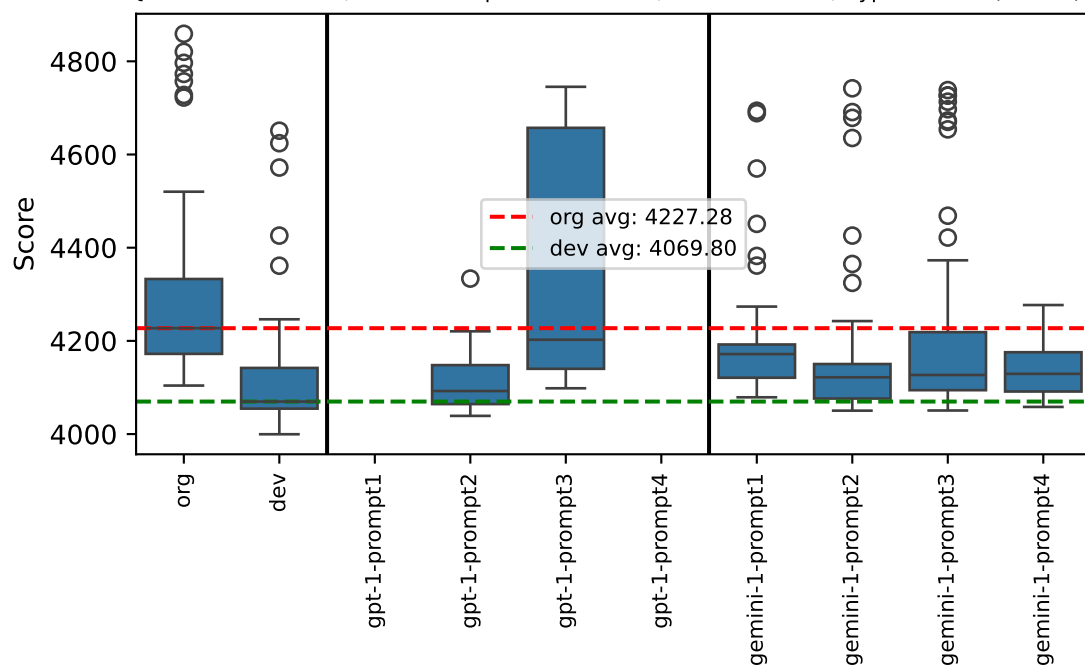
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ARRAY(BIGINT)'}



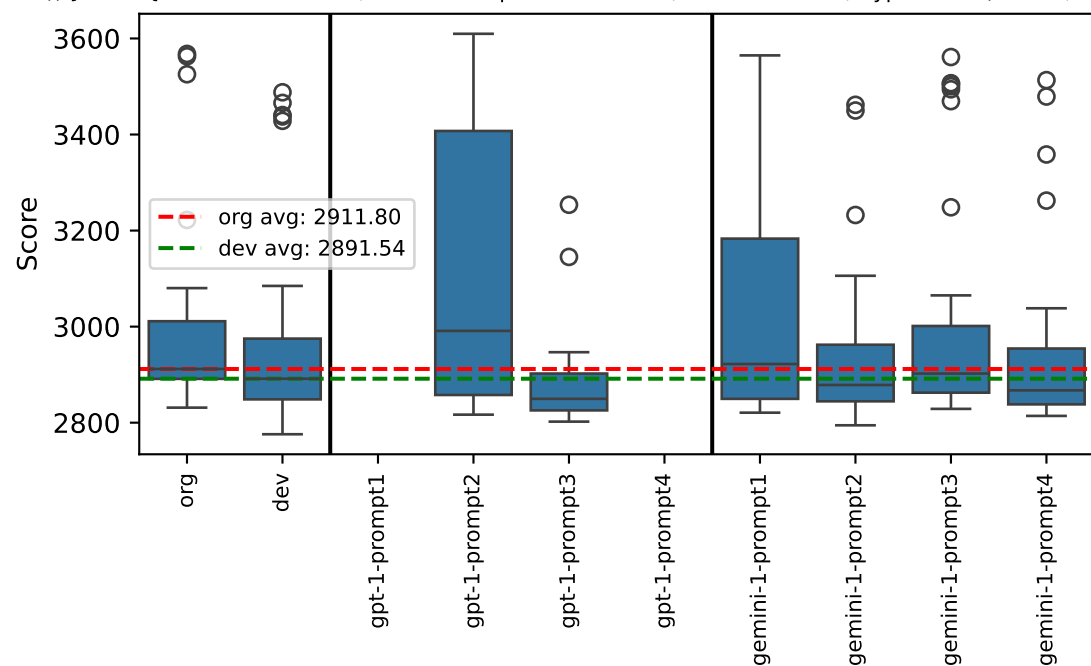
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ARRAY(VARCHAR)'}



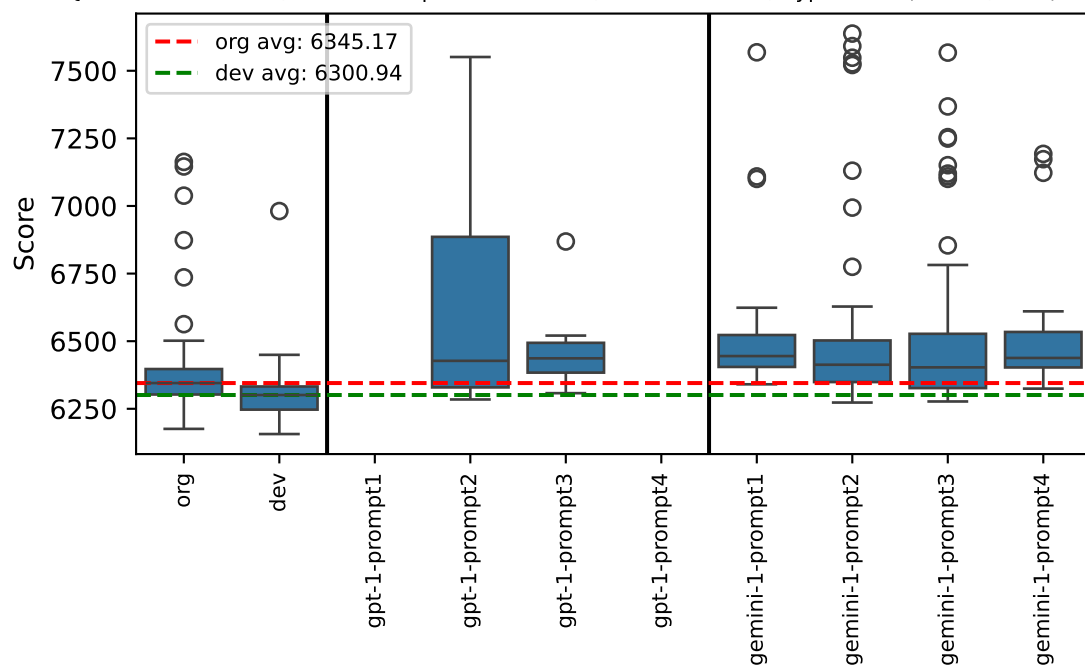
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ARRAY(ARRAY(BIGINT))'}



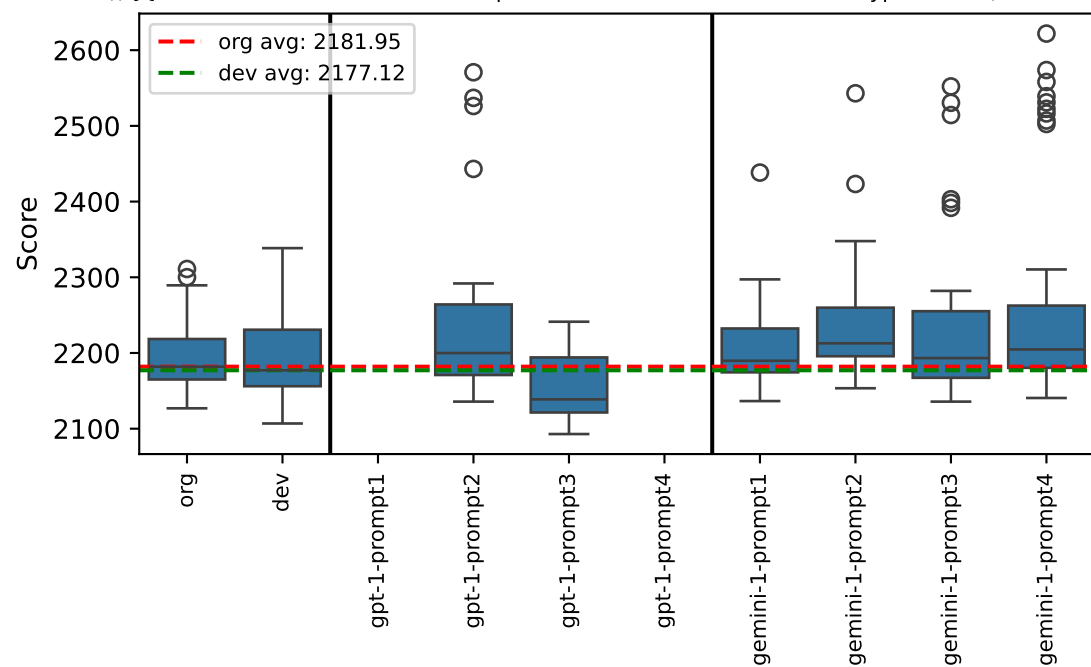
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'MAP(BIGINT, BIGINT)'}



Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'MAP(BIGINT, MAP(BIGINT, BIGINT))'}

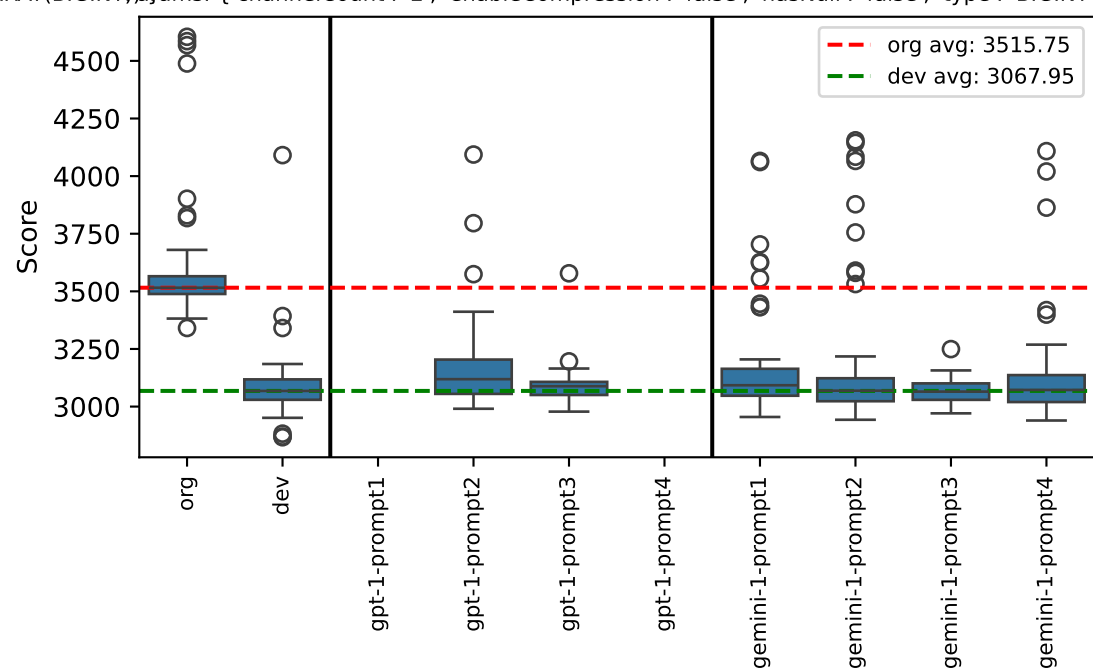
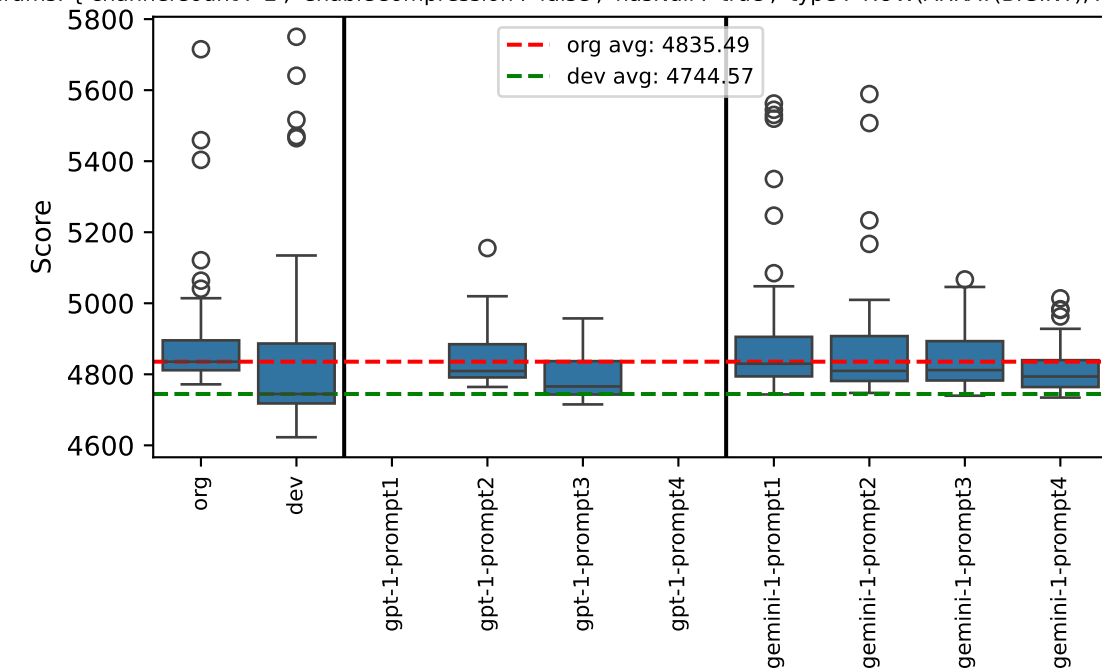


Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ROW(BIGINT, BIGINT)'}

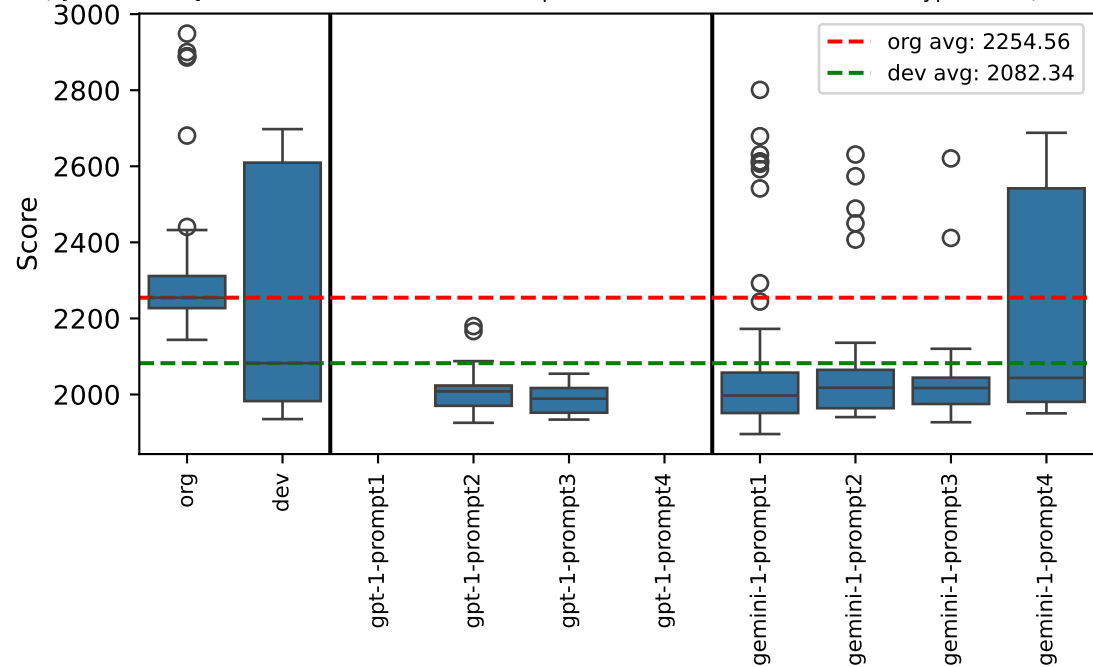
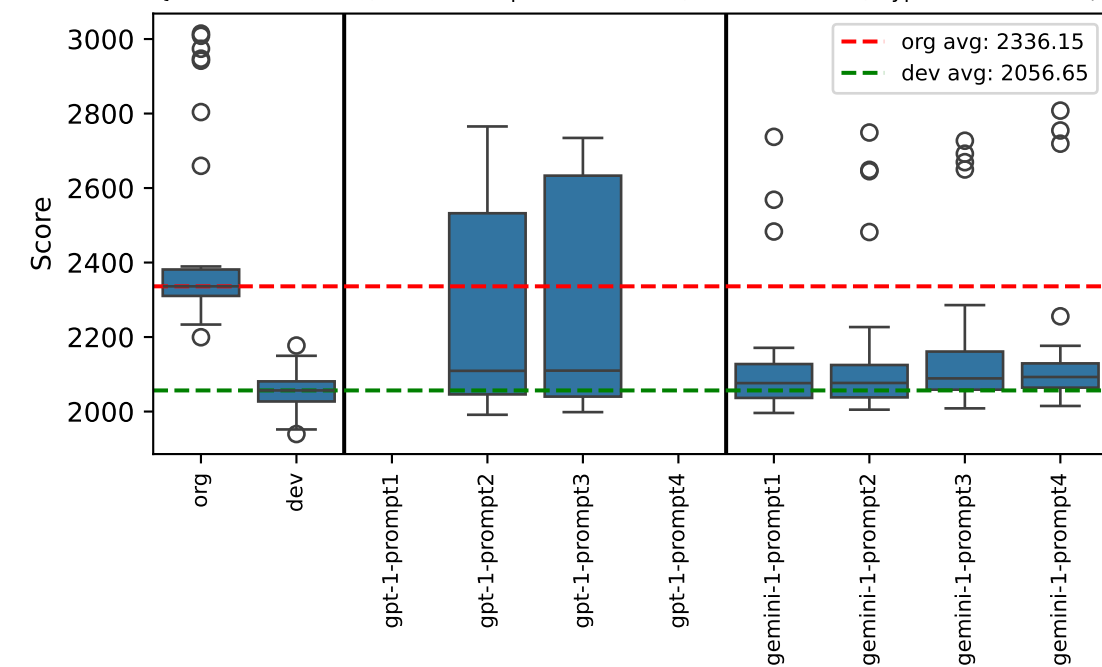


Benchmark: com.facebook.presto.operator.repartition.PartitionedOutputOperator.optimizedAddPa

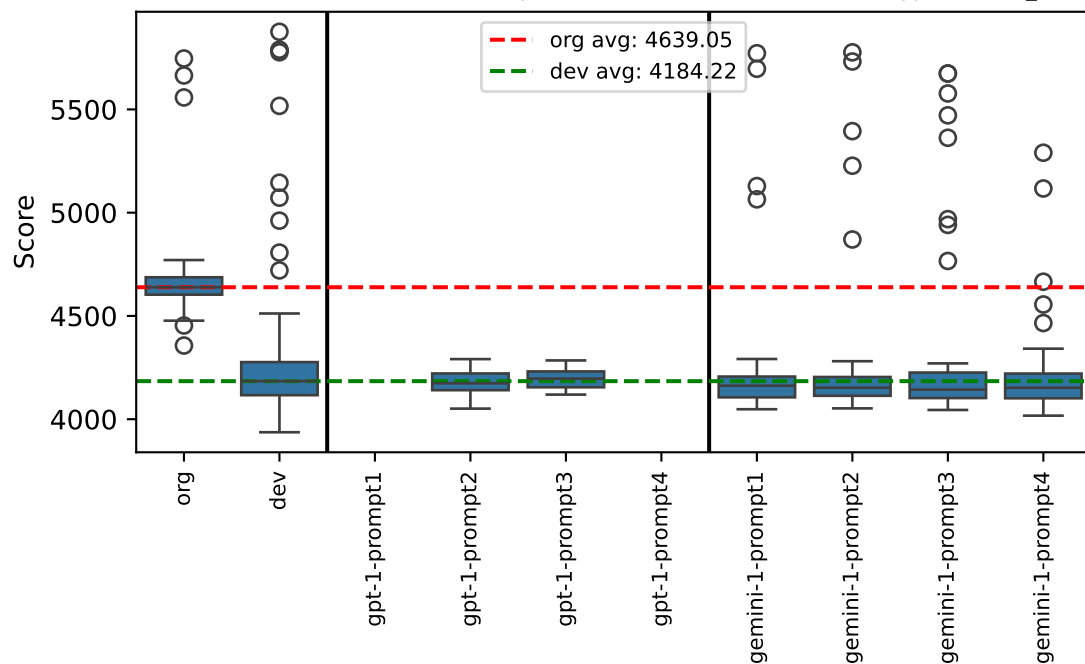
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'true', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'} Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'BIGINT'}



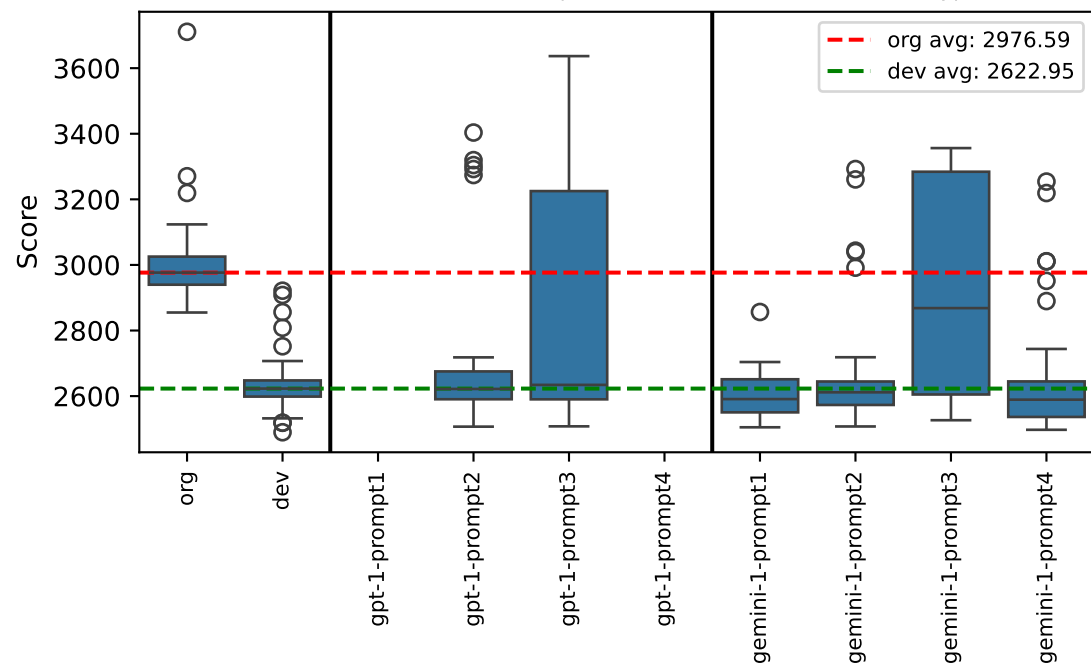
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'DICTIONARY(BIGINT)'} Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'RLE(BIGINT)'}



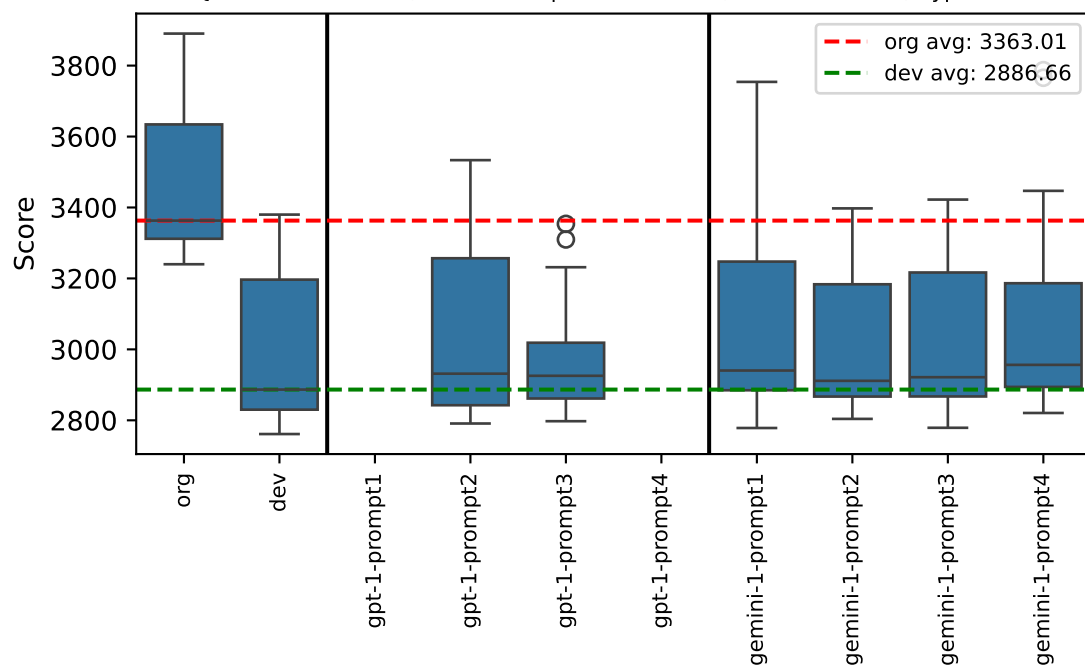
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'LONG_DECIMAL'}



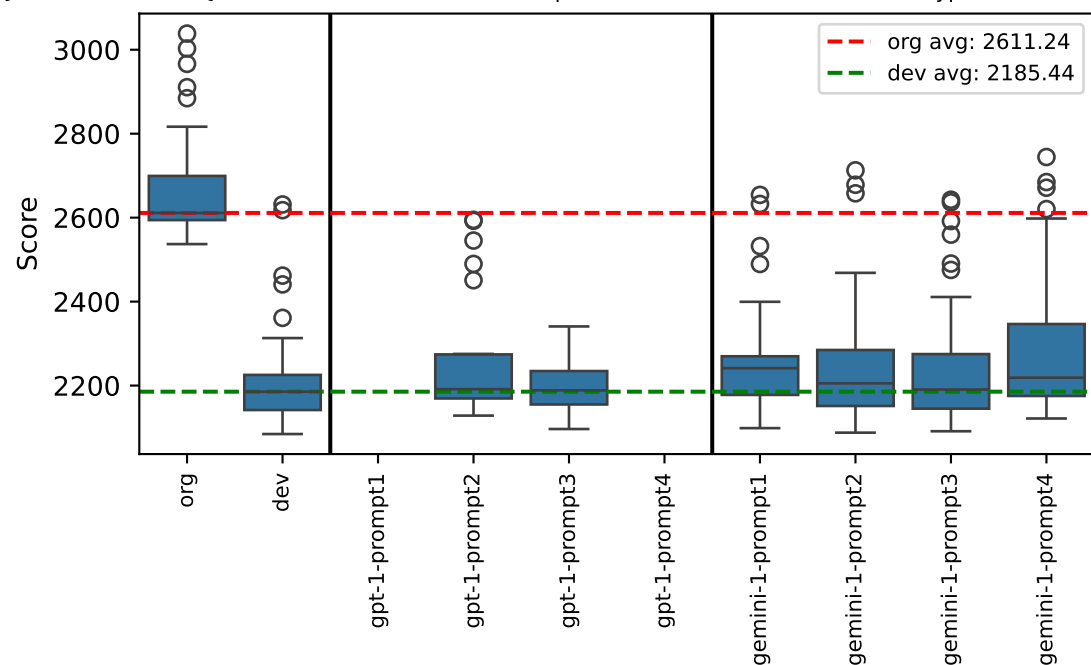
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'INTEGER'}



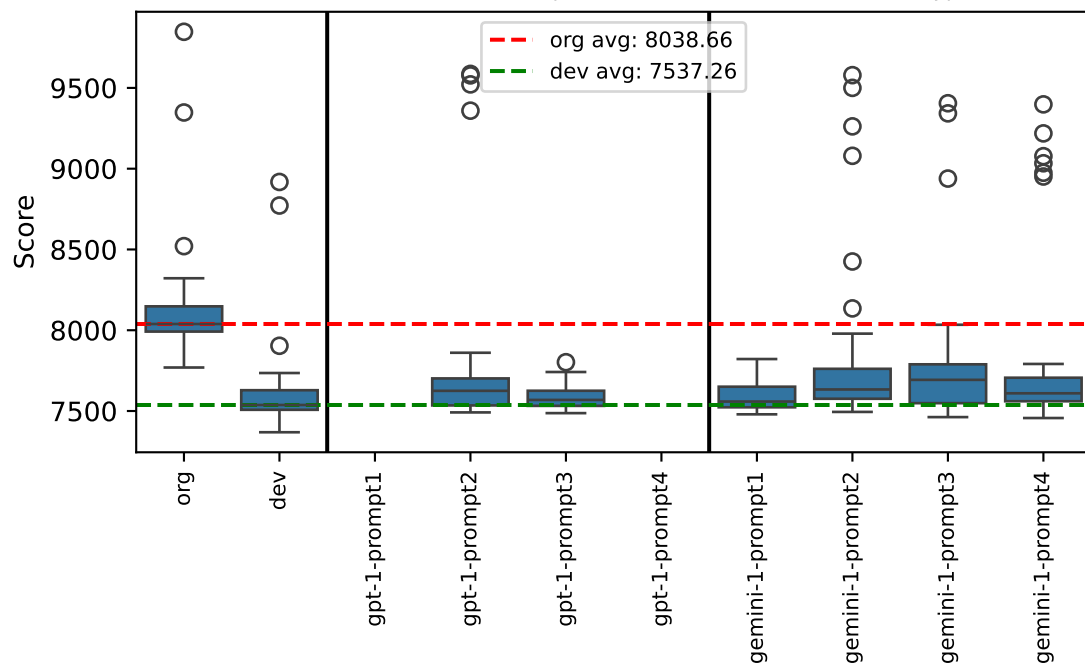
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'SMALLINT'}



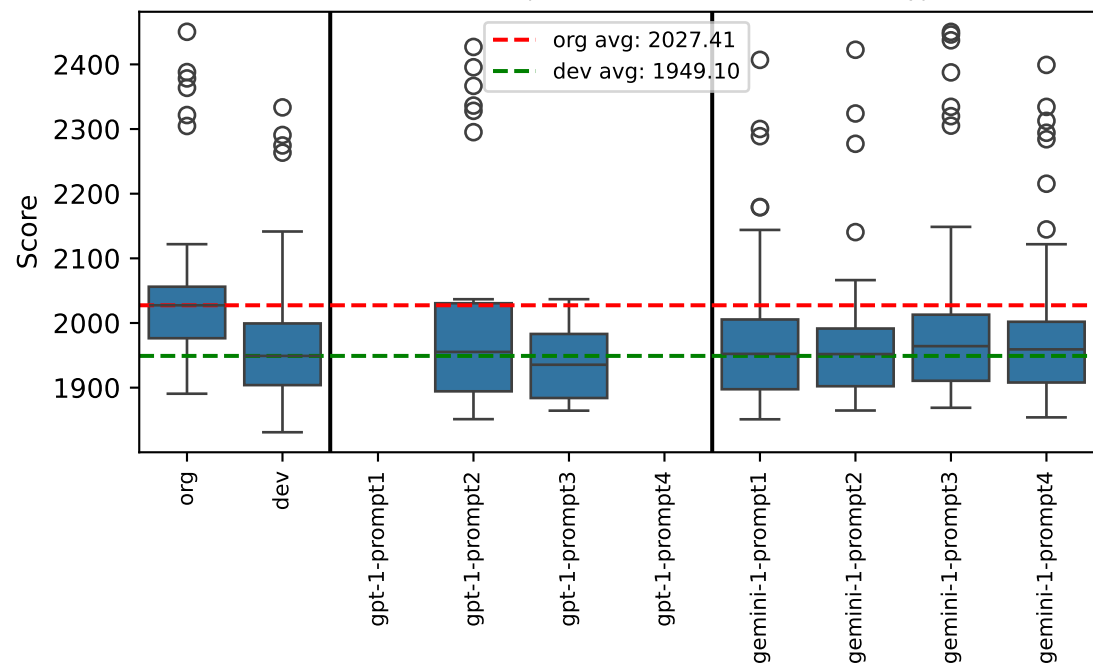
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'BOOLEAN'}



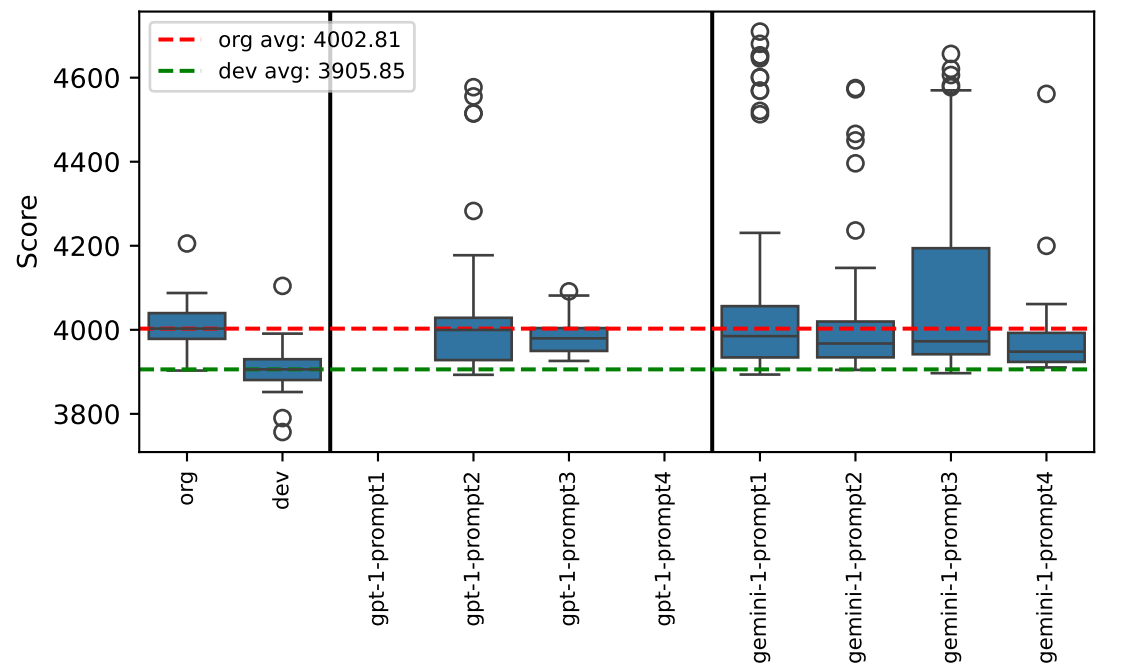
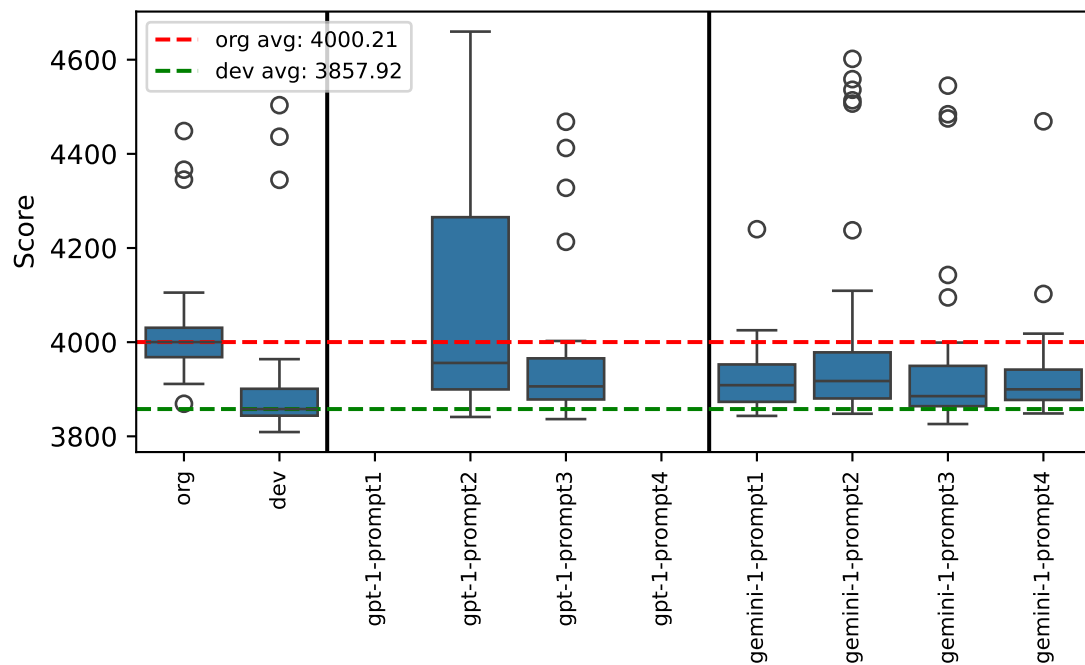
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'VARCHAR'}



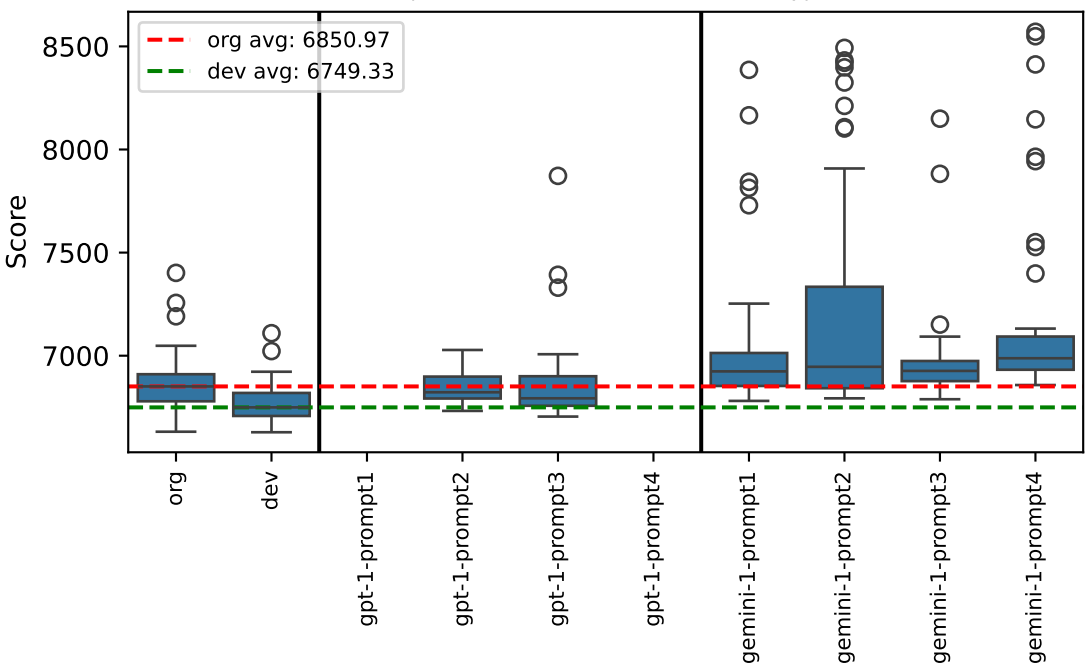
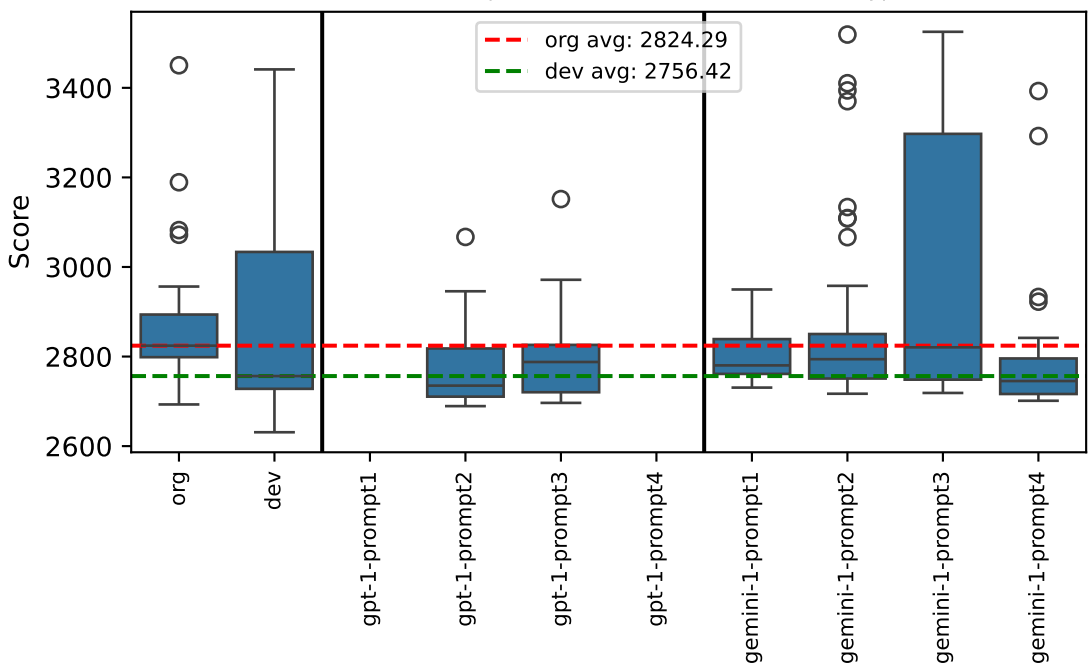
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(BIGINT



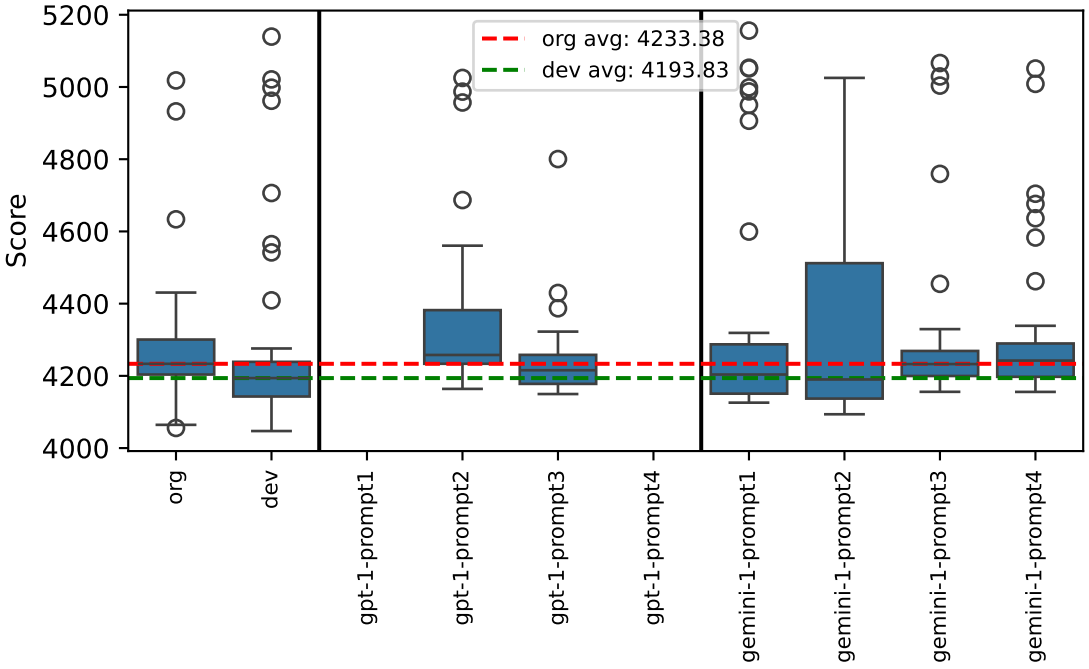
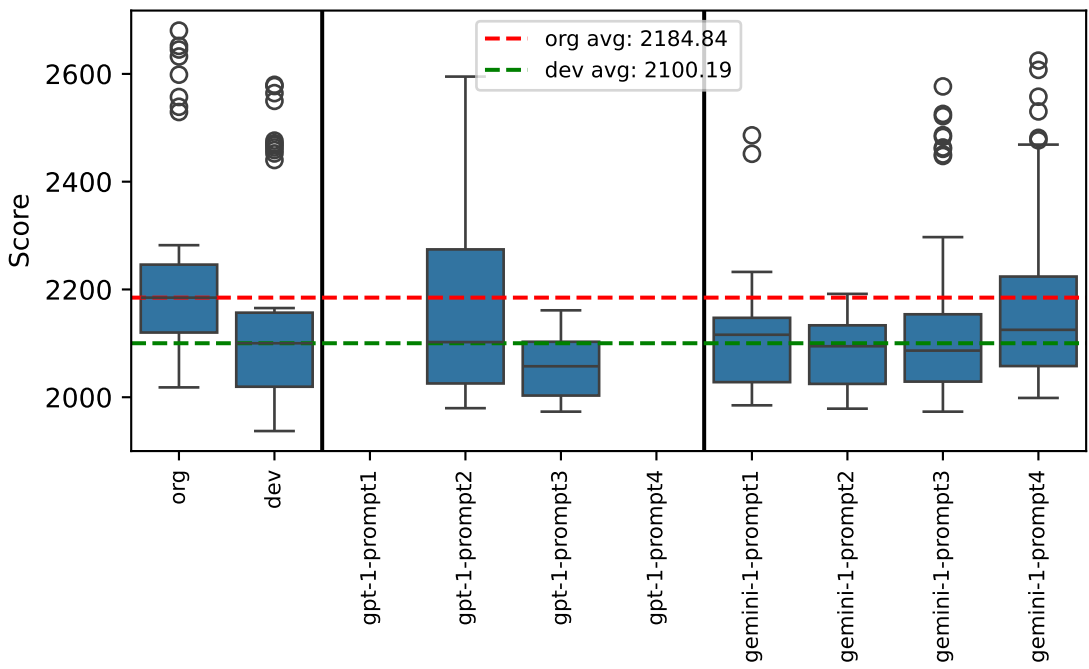
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(VARCHAR)Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ARRAY(ARRAY(BIG



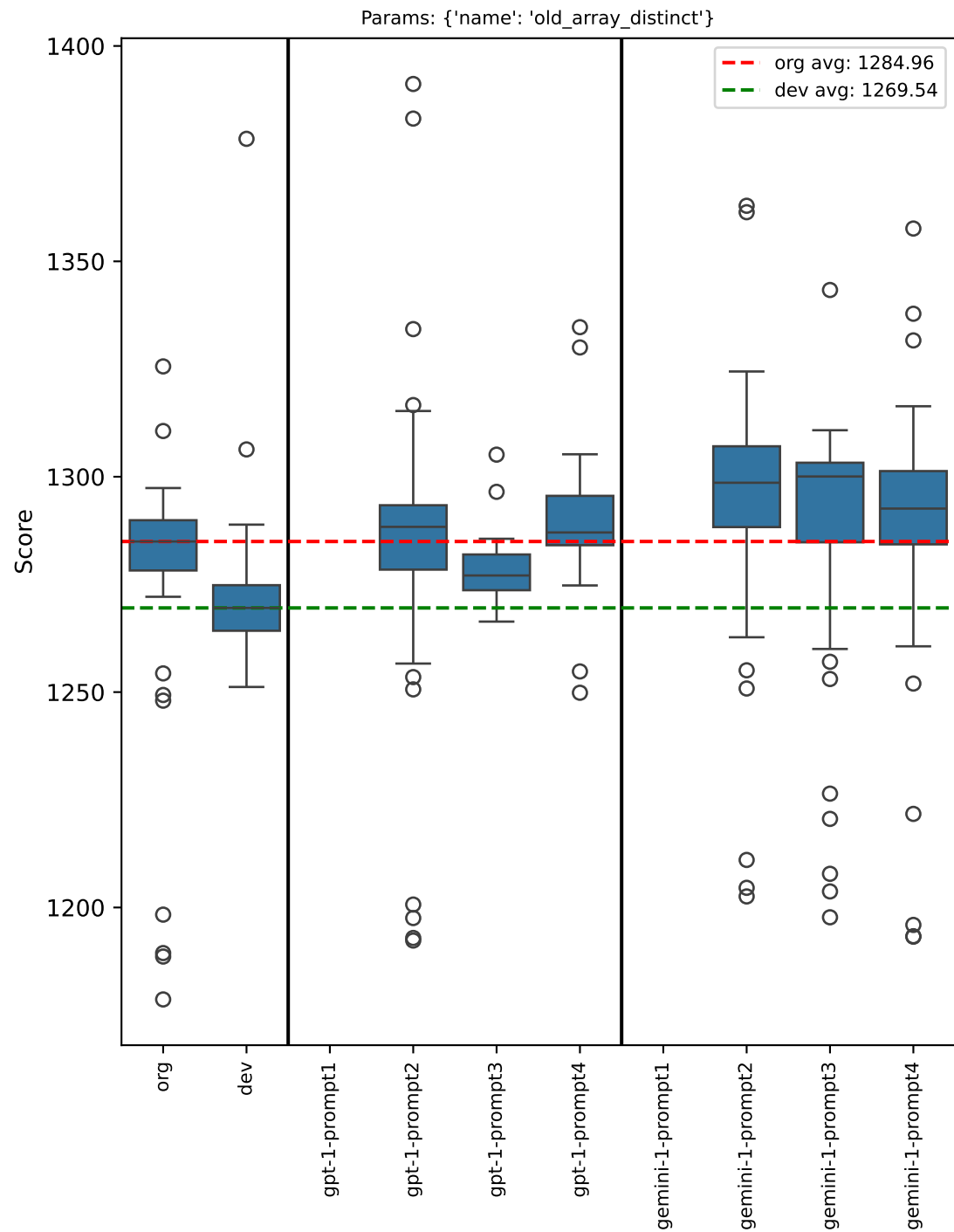
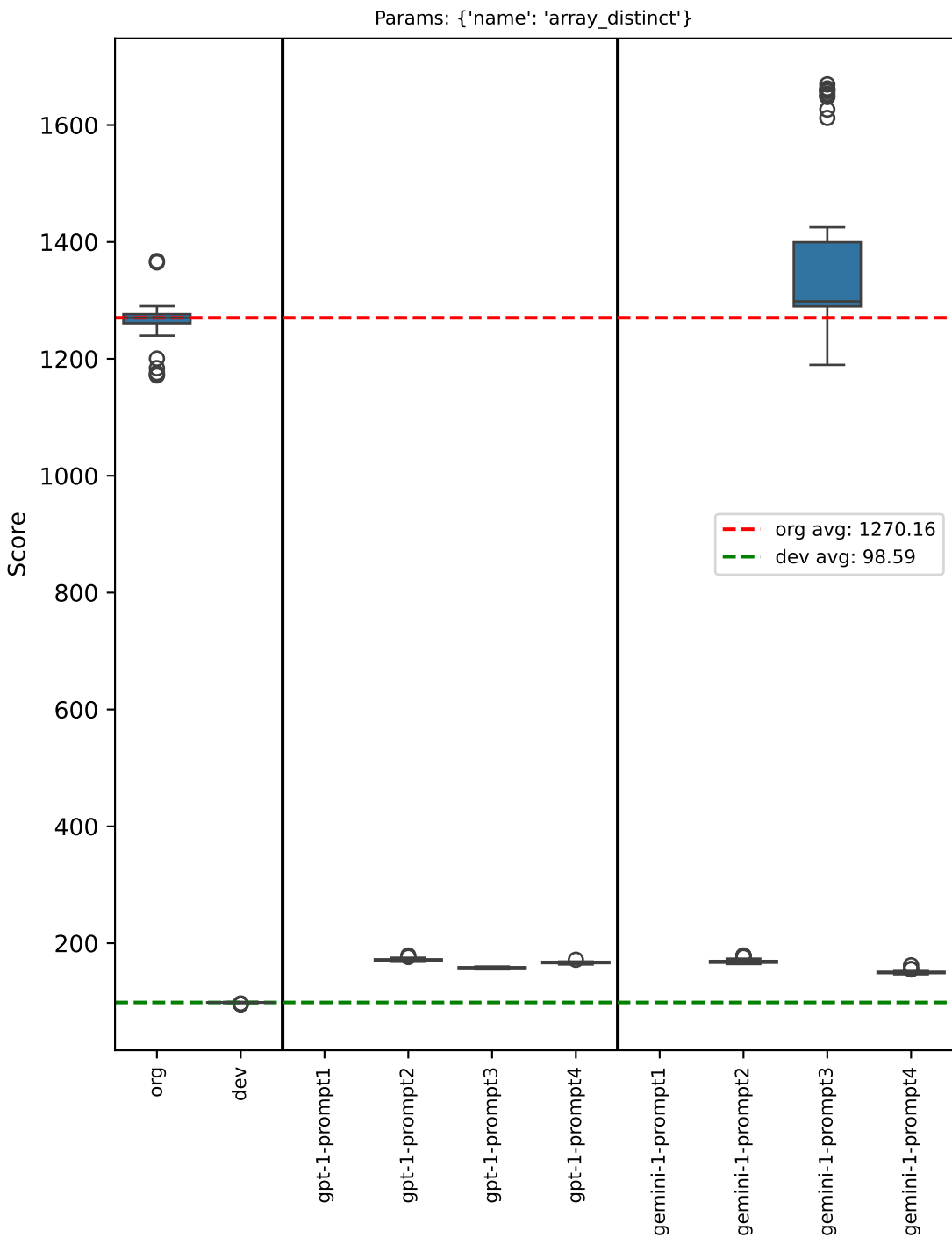
Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'MAP(BIGINT, BIGINT)'} Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'MAP(BIGINT, MAP(BIGINT, BIGINT))'}



Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ROW(BIGINT, BIGINT)'} Params: {'channelCount': '2', 'enableCompression': 'false', 'hasNull': 'false', 'type': 'ROW(ARRAY(BIGINT), ARRAY(BIGINT))'}

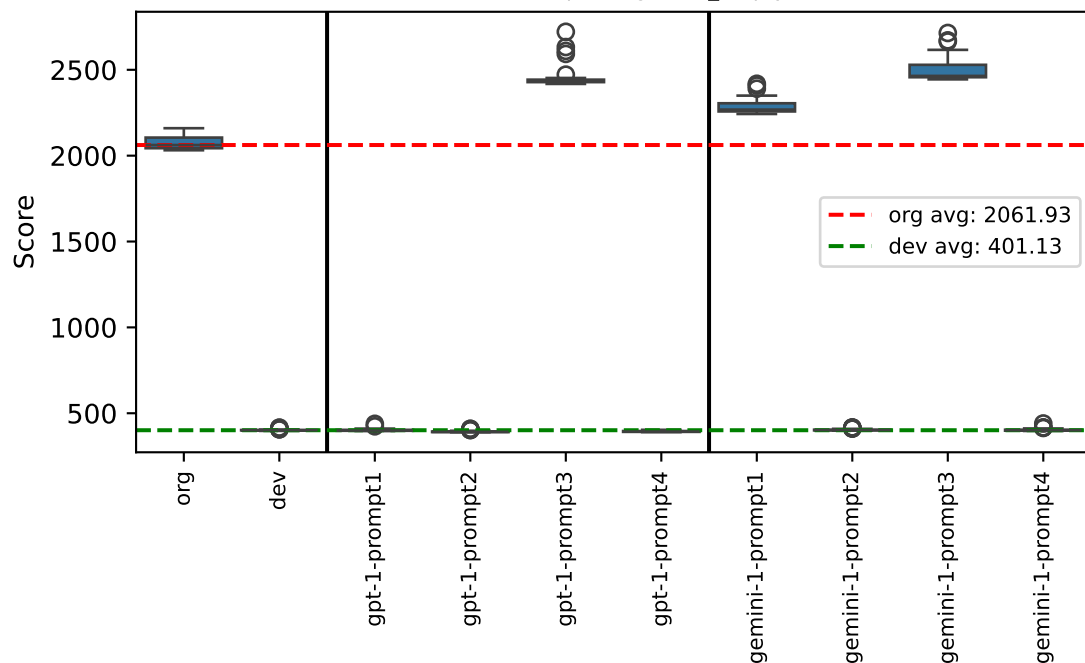


Benchmark: com.facebook.presto.operator.scalar.BenchmarkArrayDistinct.arrayDistinct

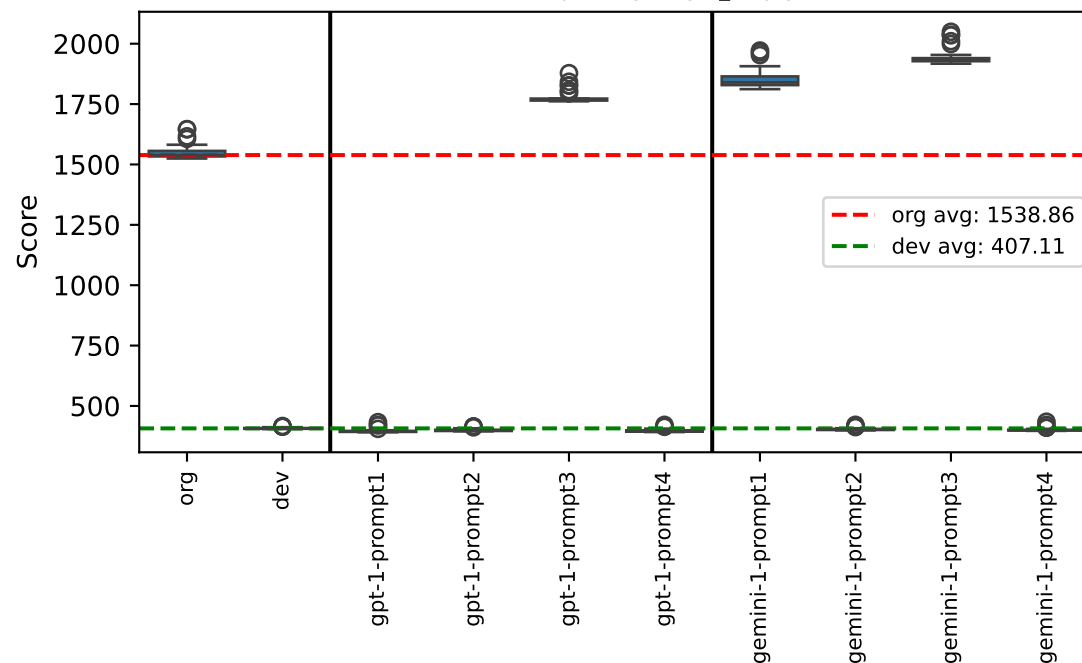


Benchmark: com.facebook.presto.operator.scalar.BenchmarkMapConcat.mapConcat

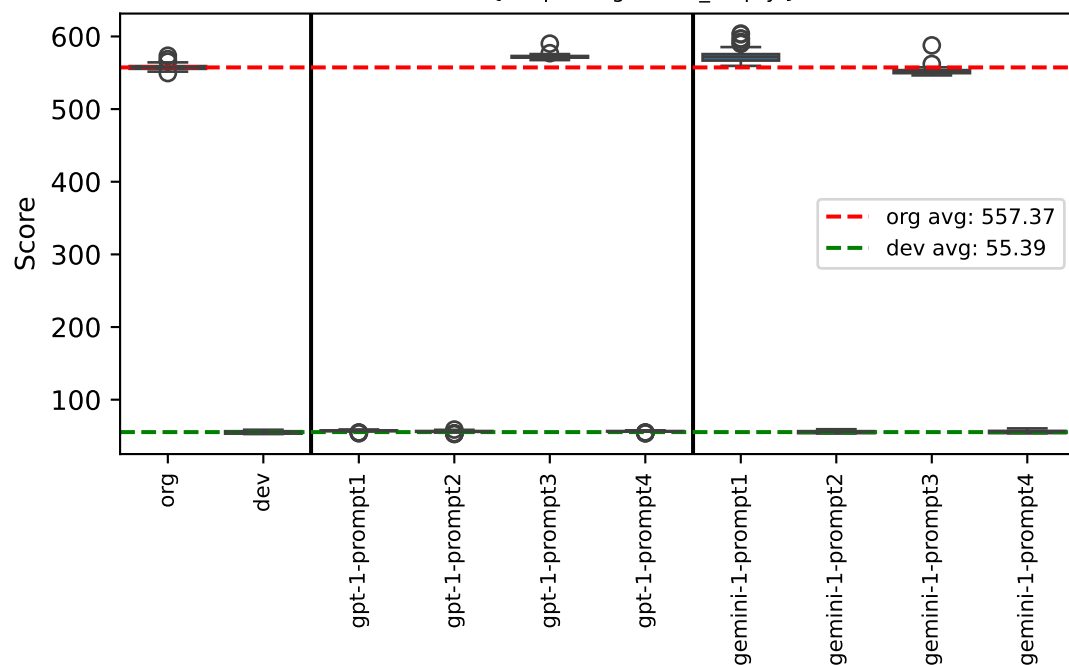
Params: {'mapConfig': 'left_empty'}



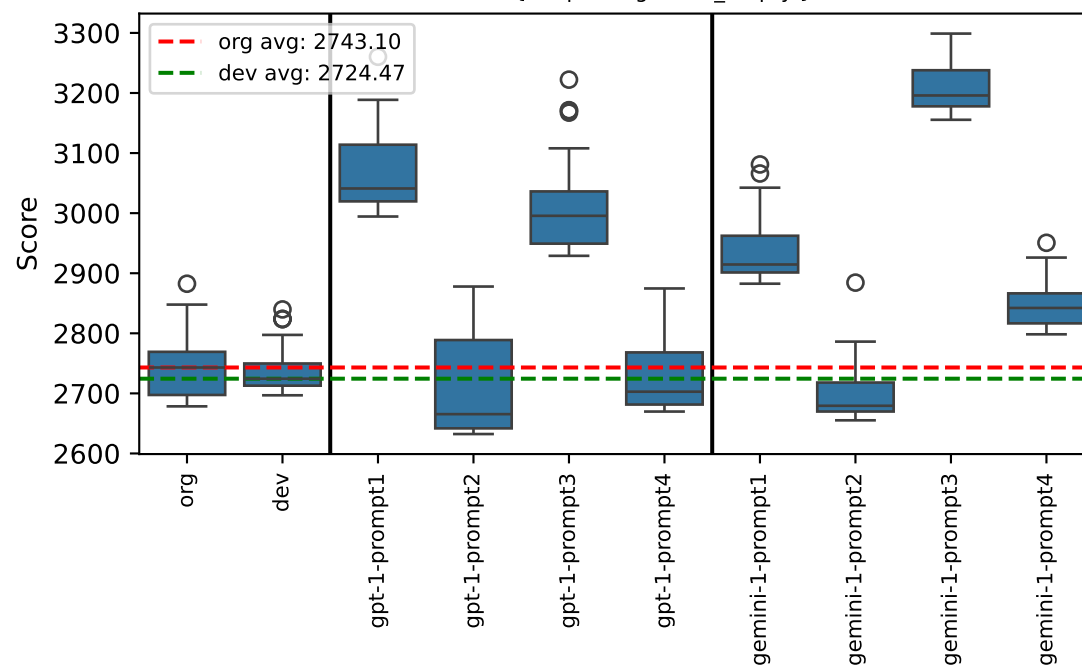
Params: {'mapConfig': 'right_empty'}



Params: {'mapConfig': 'both_empty'}

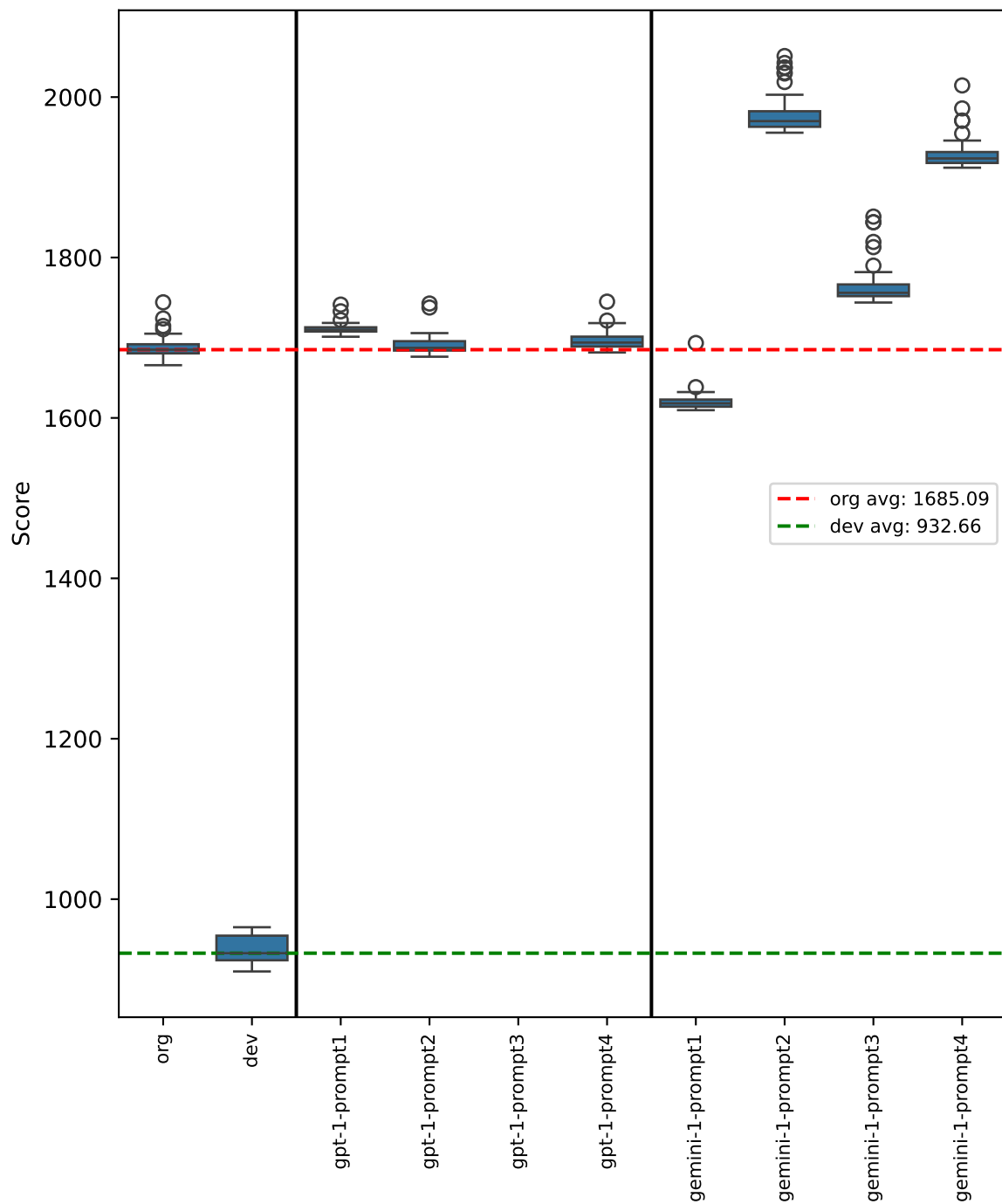


Params: {'mapConfig': 'non_empty'}

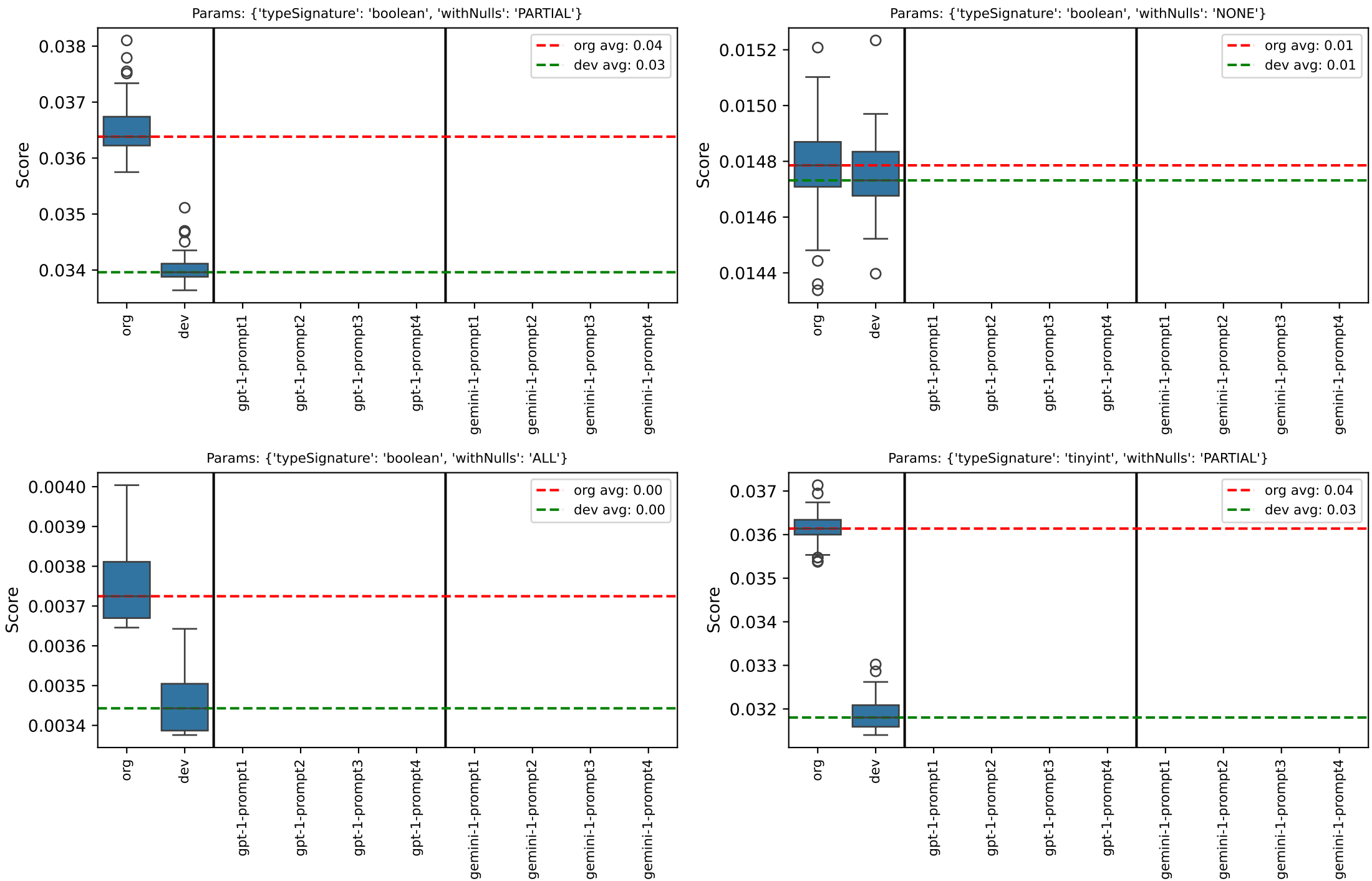


Benchmark: com.facebook.presto.operator.scalar.BenchmarkMapToMapCast.benchmark

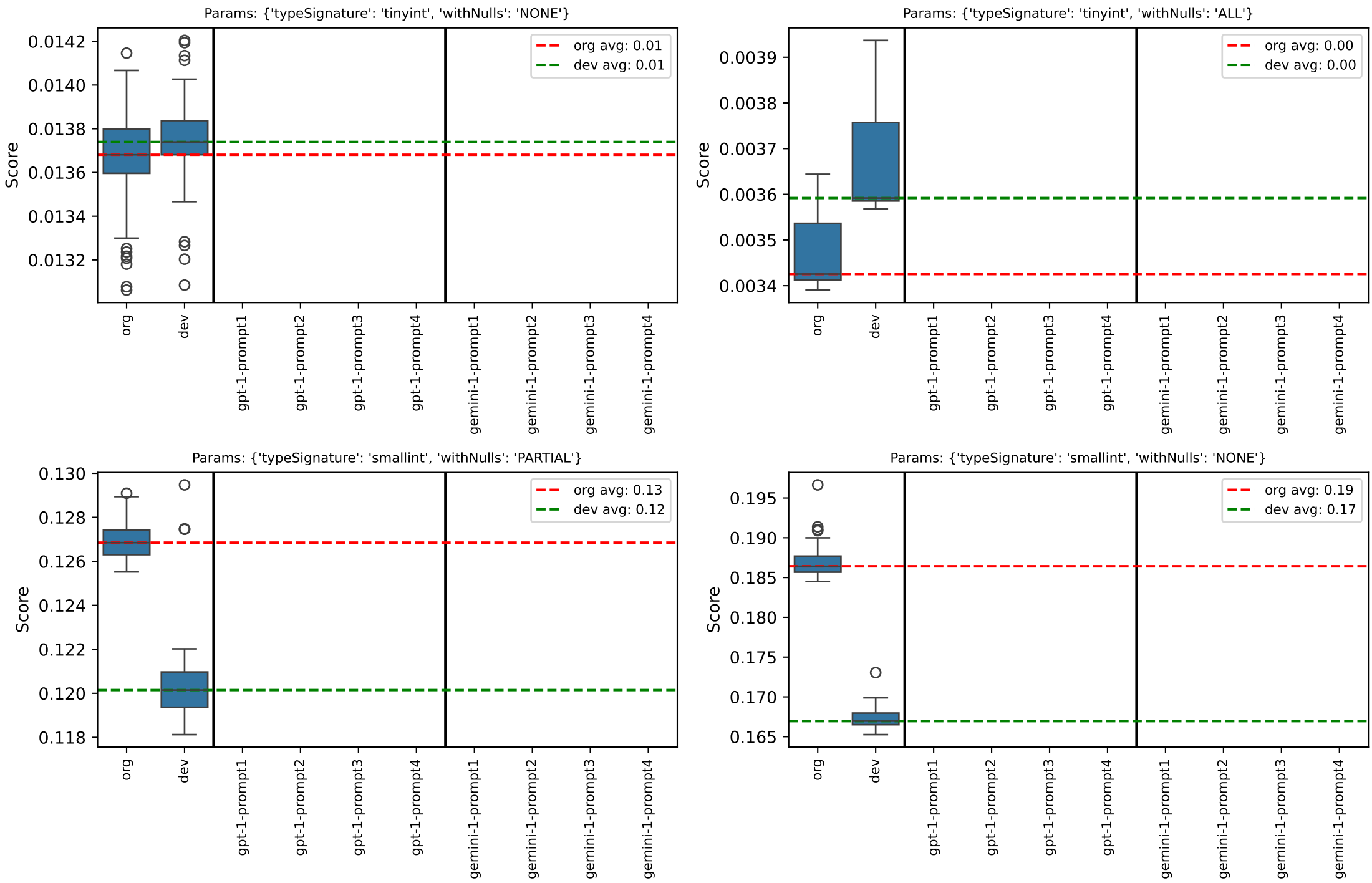
Params: {}



Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

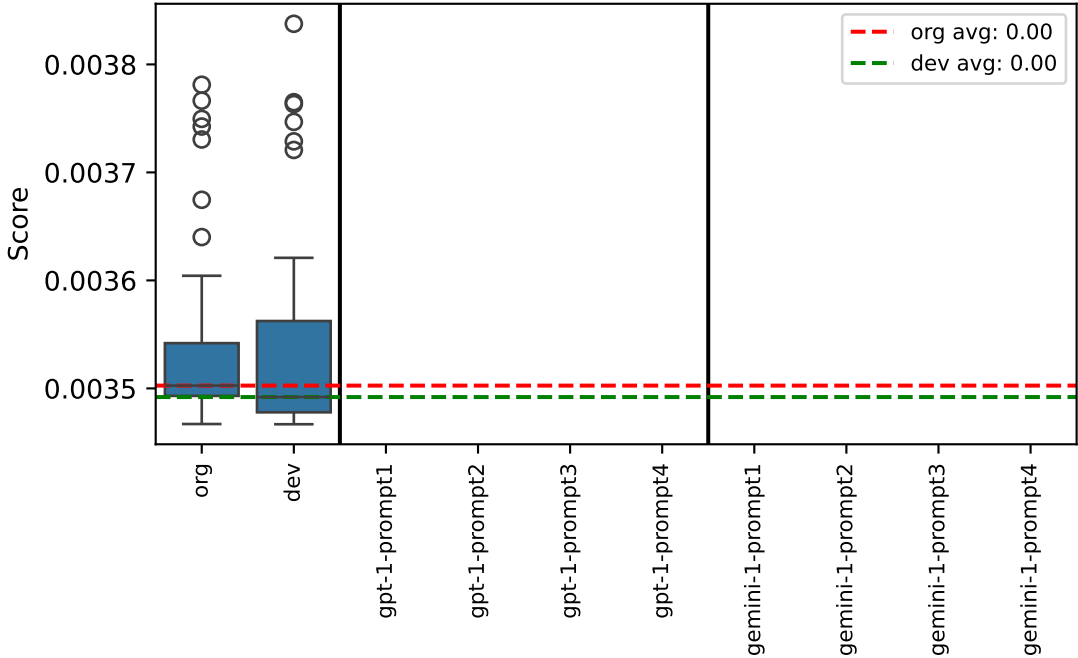


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

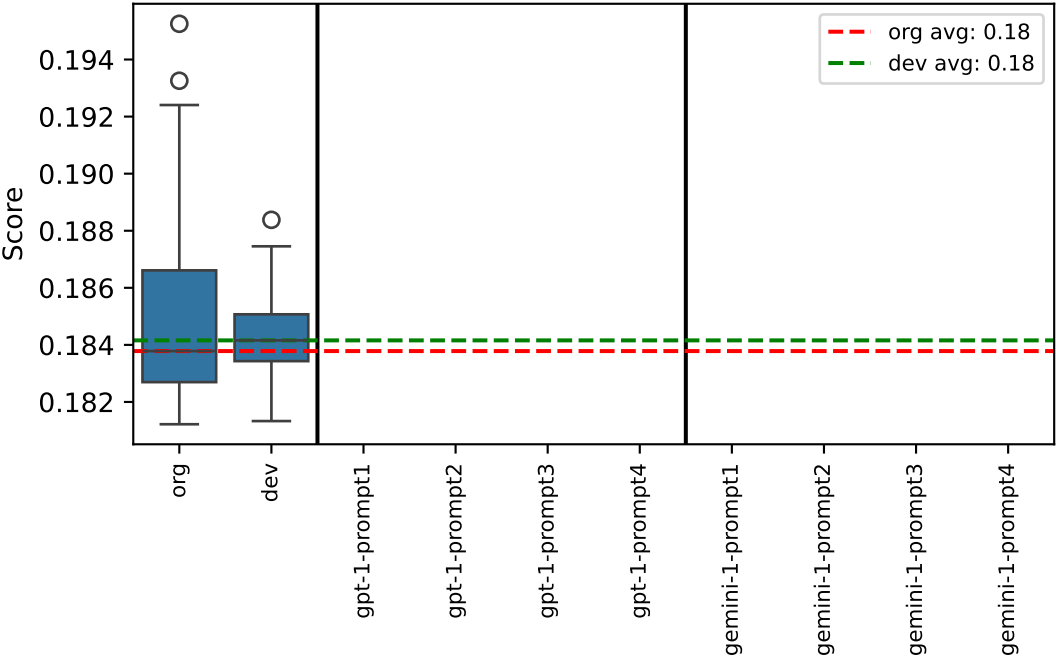


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

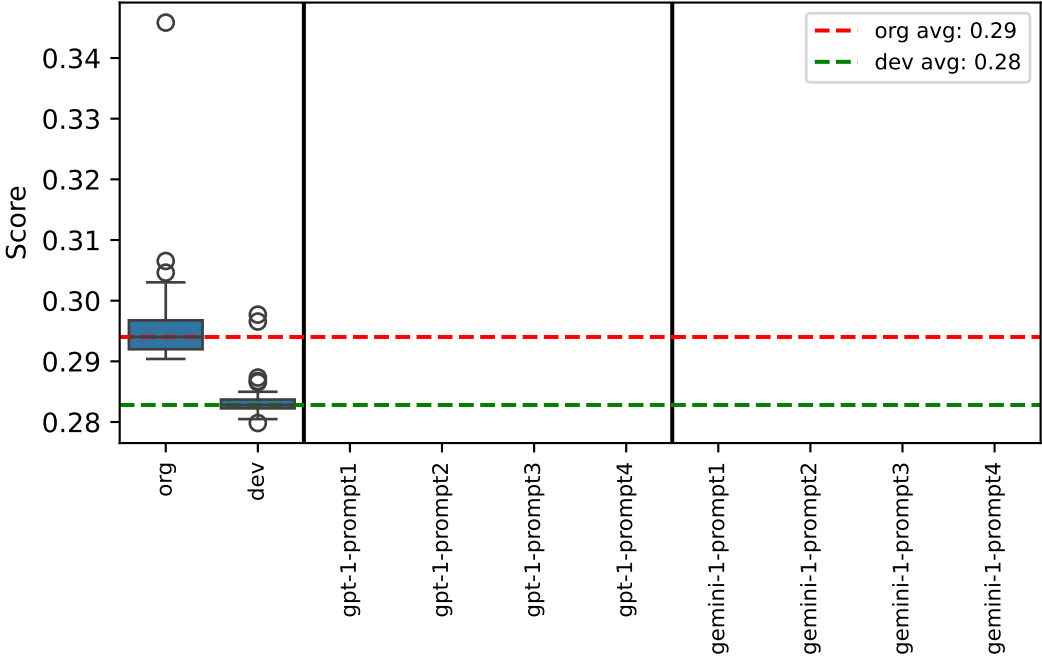
Params: {'typeSignature': 'smallint', 'withNulls': 'ALL'}



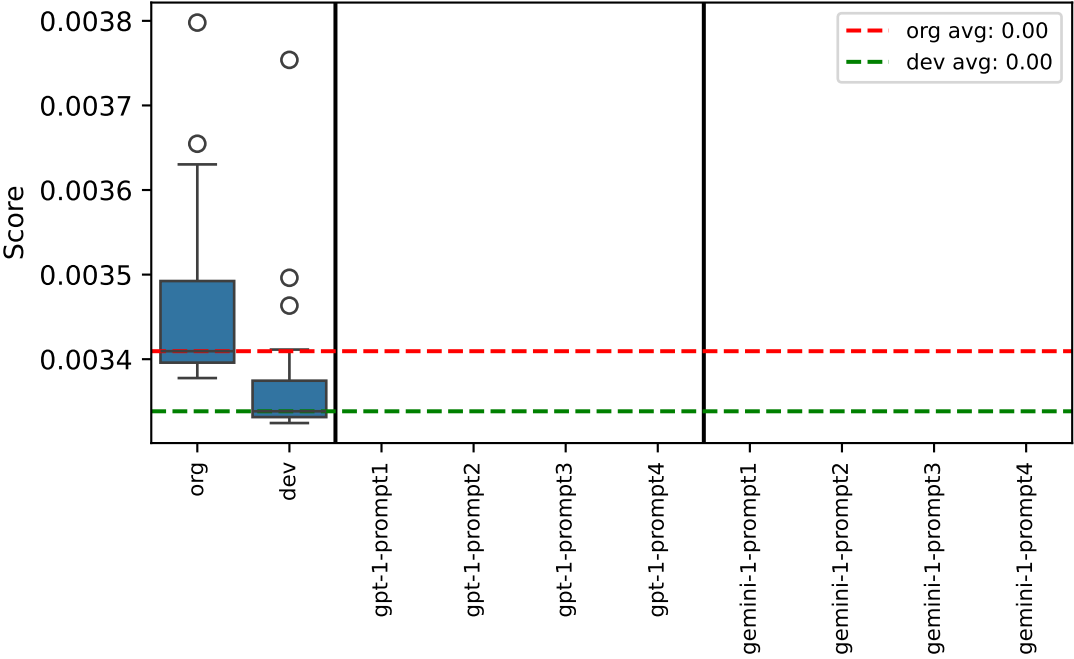
Params: {'typeSignature': 'integer', 'withNulls': 'PARTIAL'}



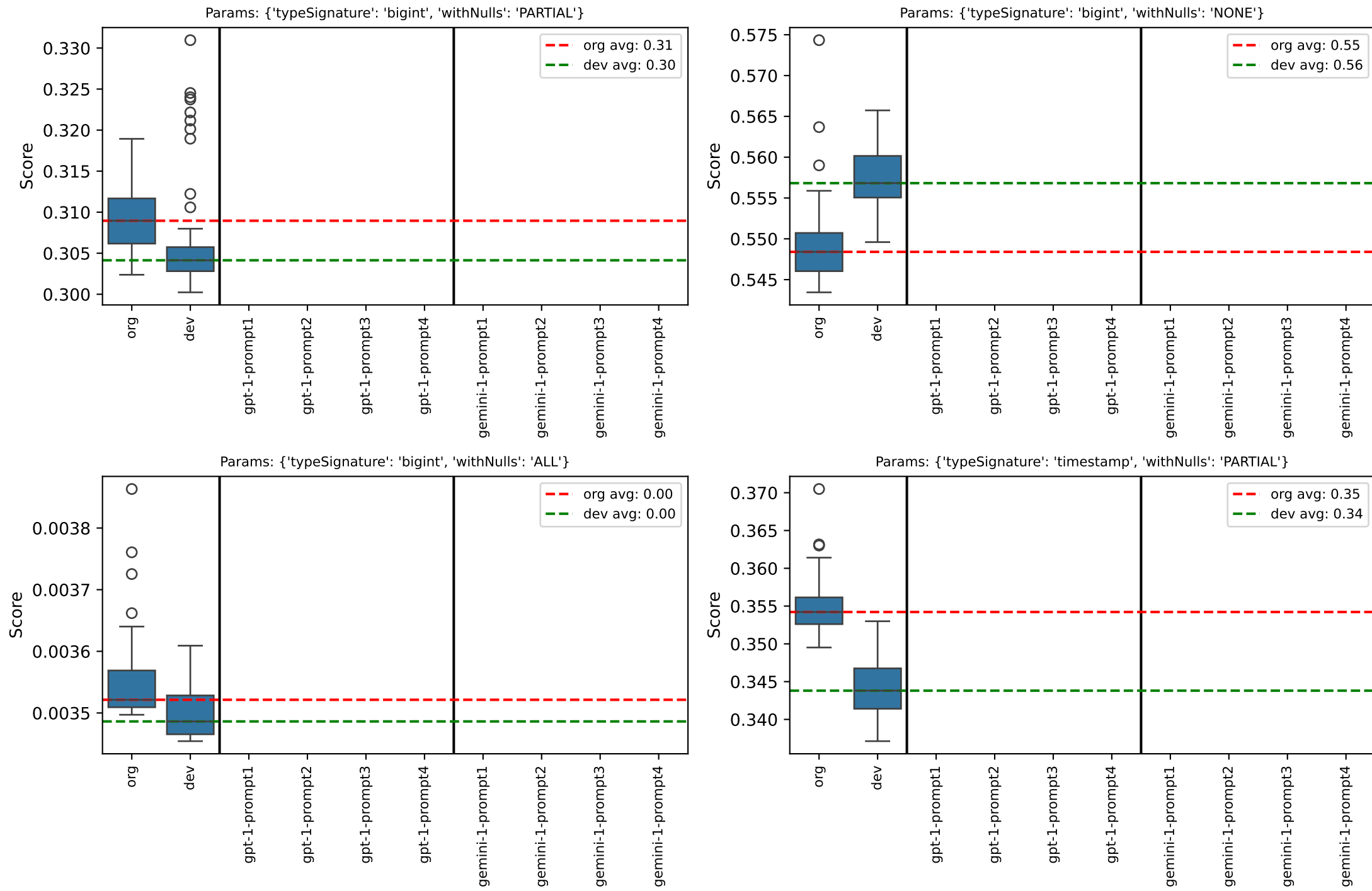
Params: {'typeSignature': 'integer', 'withNulls': 'NONE'}



Params: {'typeSignature': 'integer', 'withNulls': 'ALL'}

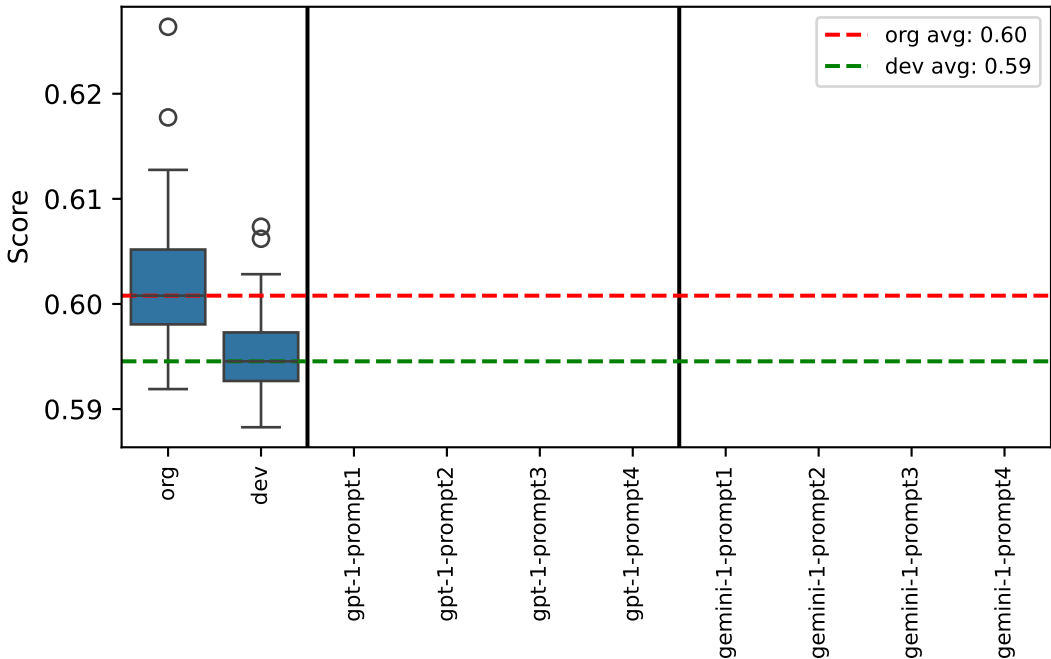


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

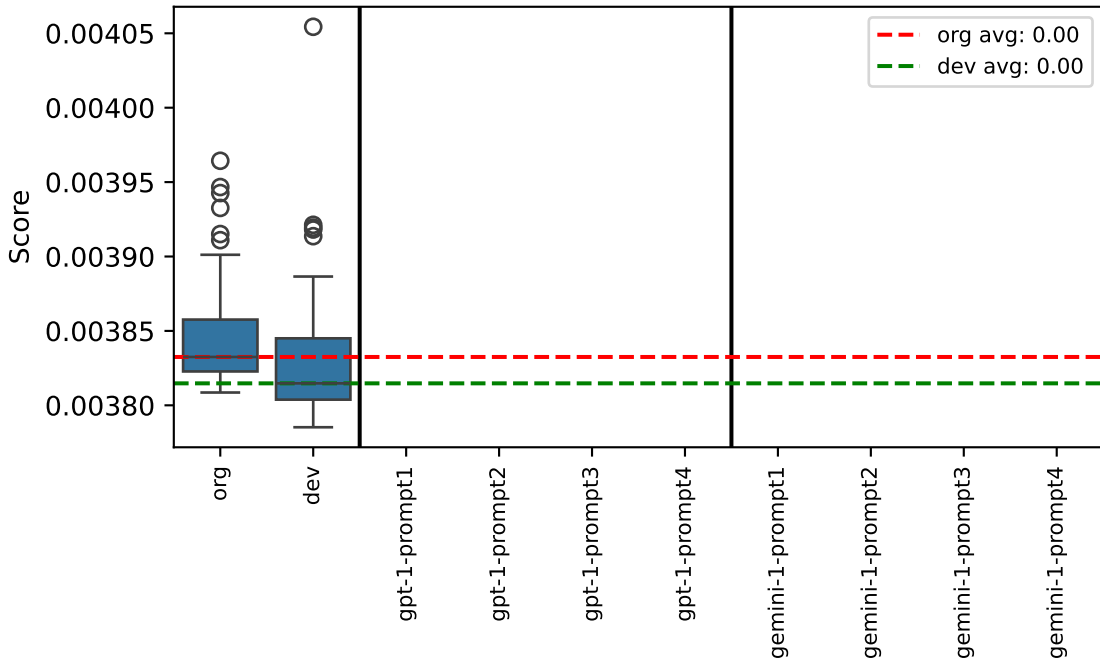


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

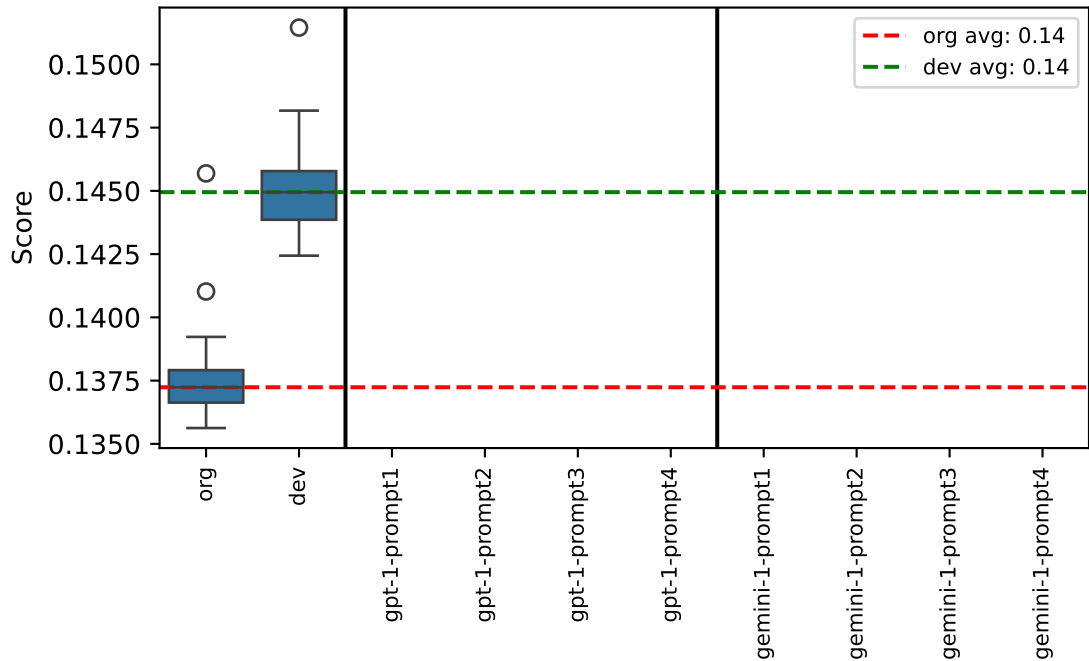
Params: {'typeSignature': 'timestamp', 'withNulls': 'NONE'}



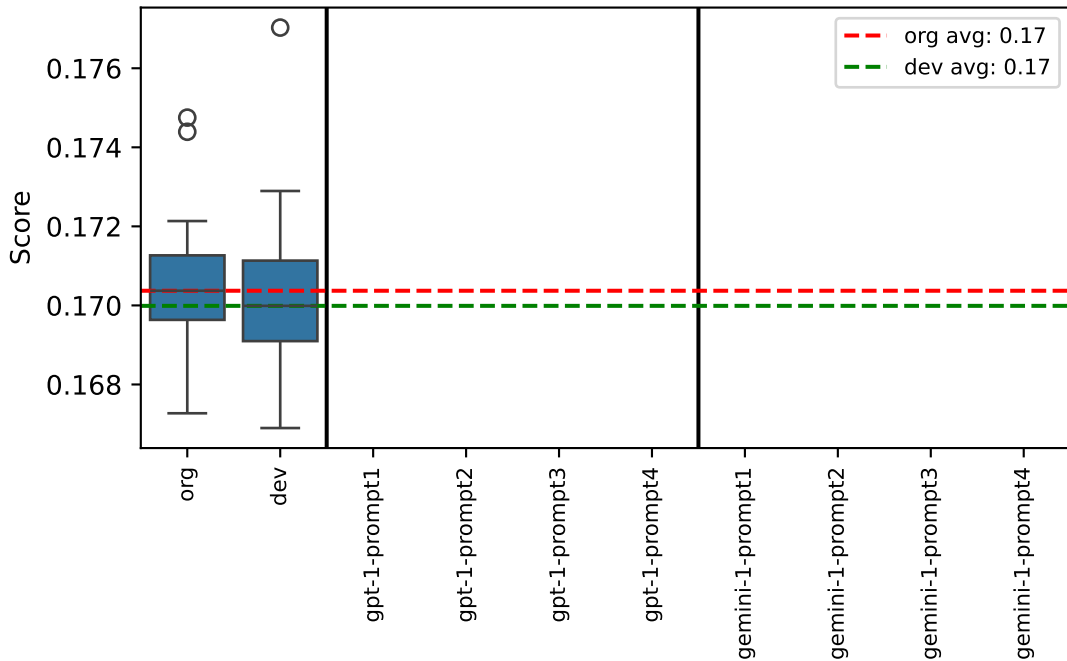
Params: {'typeSignature': 'timestamp', 'withNulls': 'ALL'}



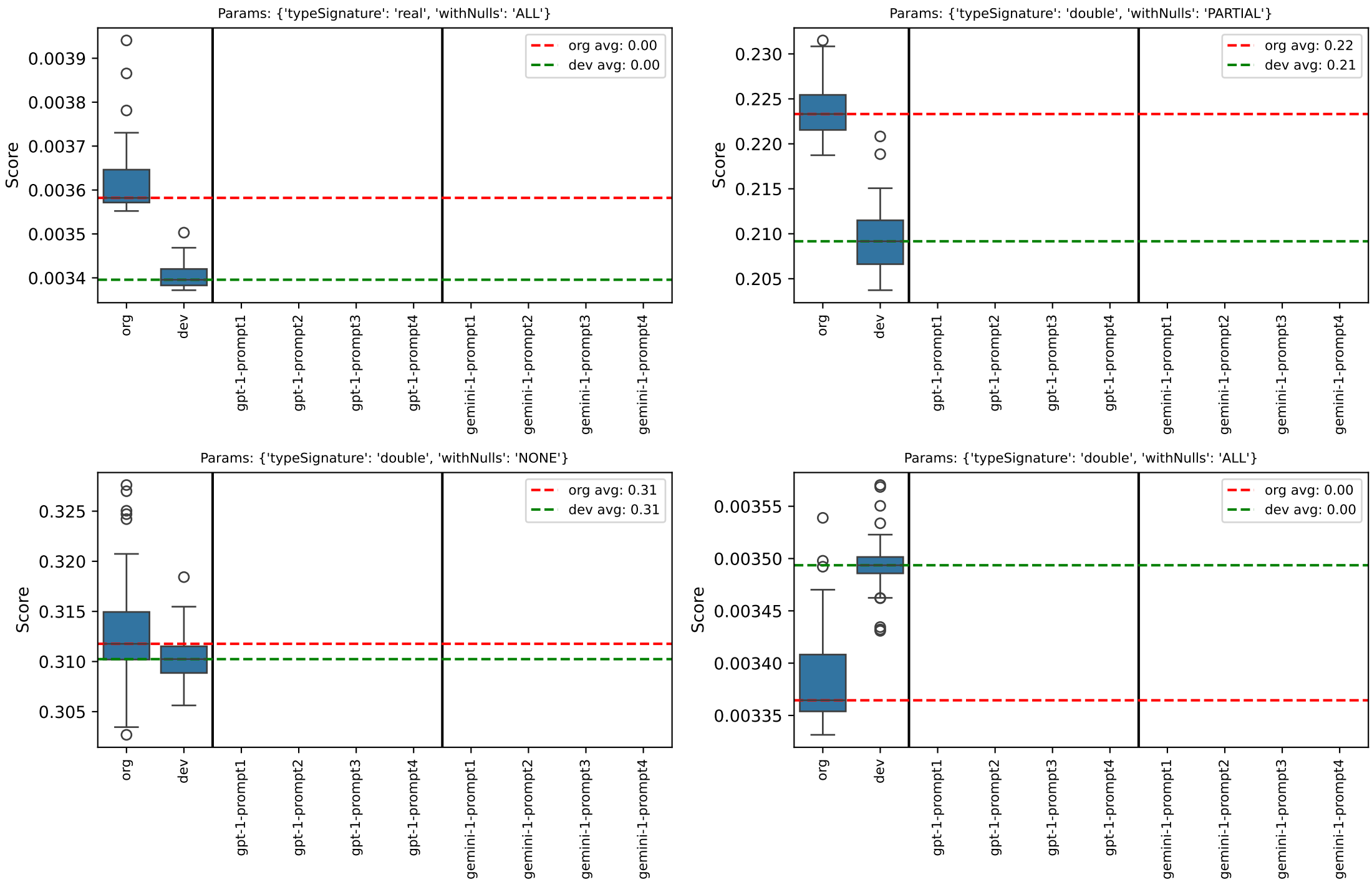
Params: {'typeSignature': 'real', 'withNulls': 'PARTIAL'}



Params: {'typeSignature': 'real', 'withNulls': 'NONE'}

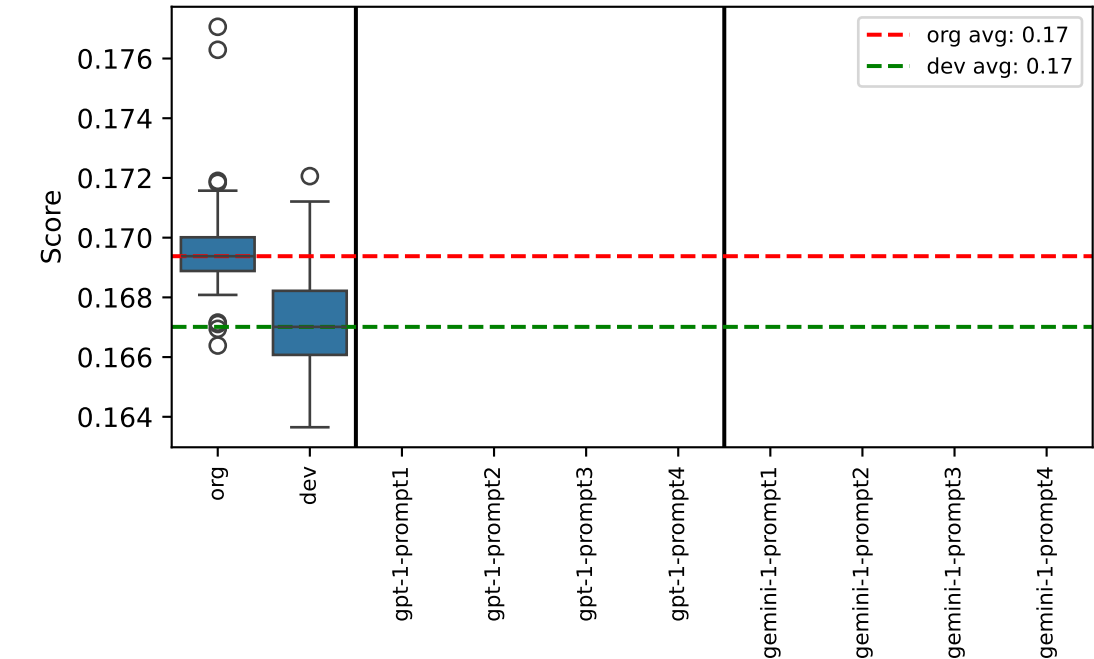


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

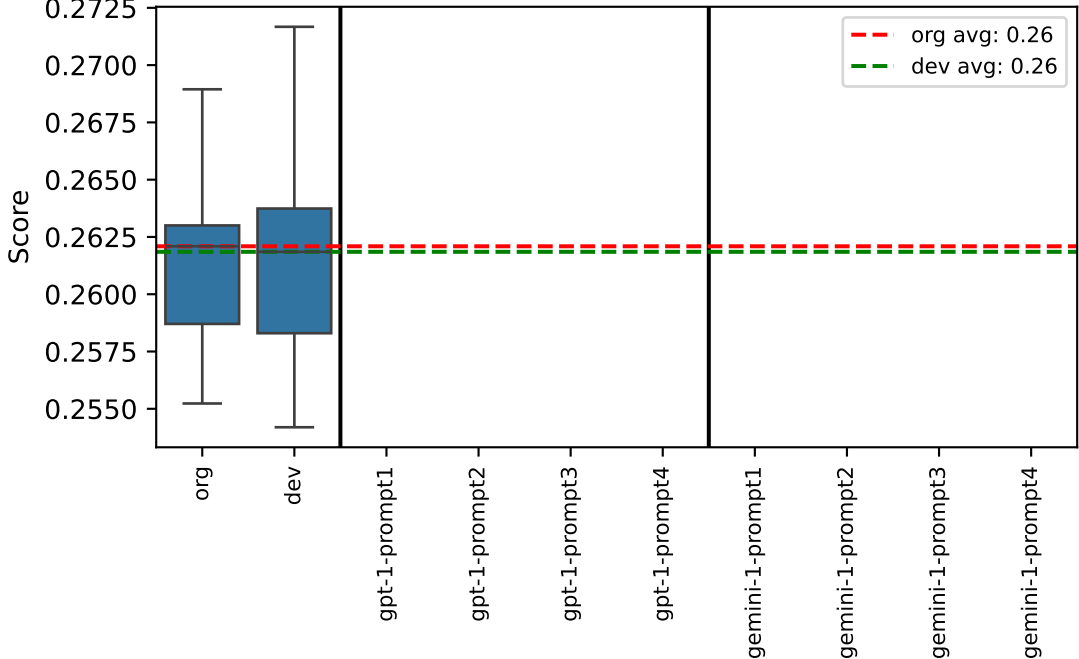


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithJni

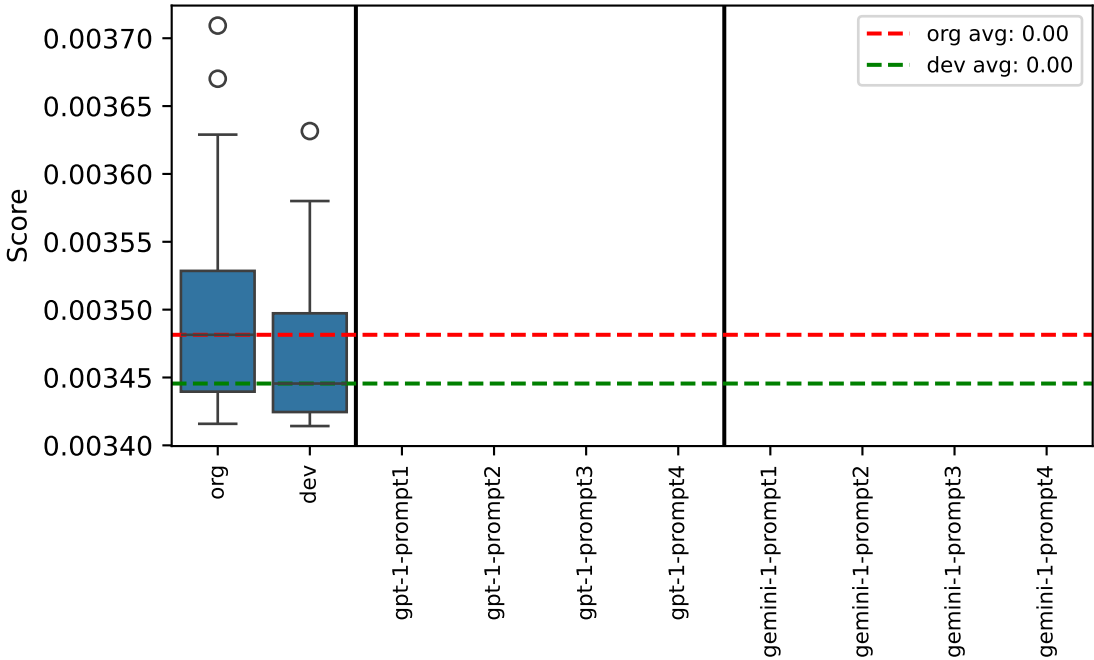
Params: {'typeSignature': 'varchar_direct', 'withNulls': 'PARTIAL'}



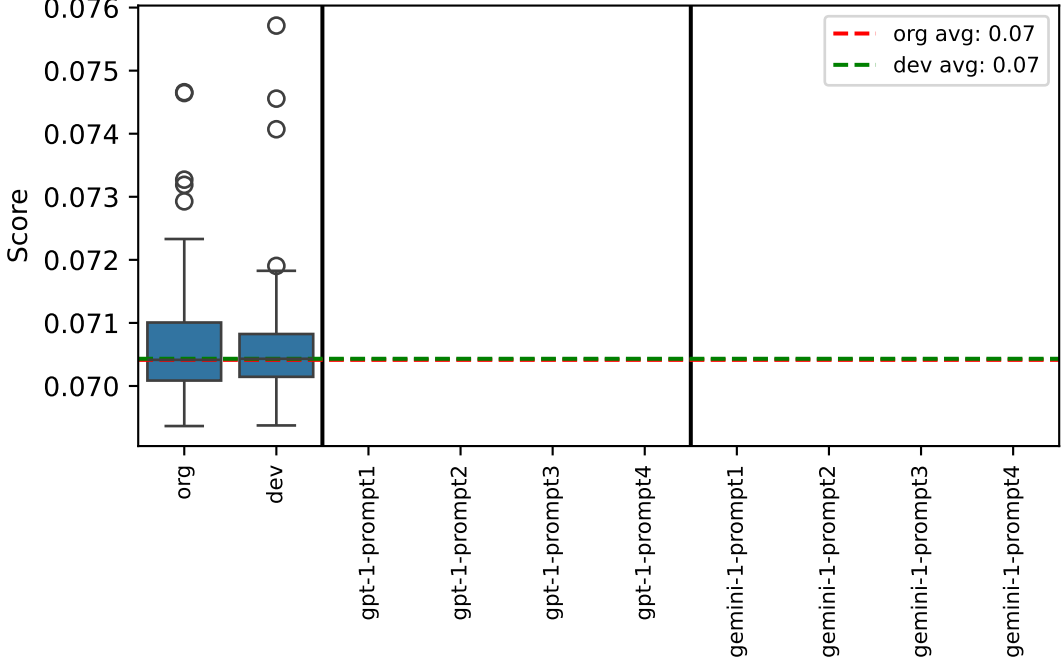
Params: {'typeSignature': 'varchar_direct', 'withNulls': 'NONE'}



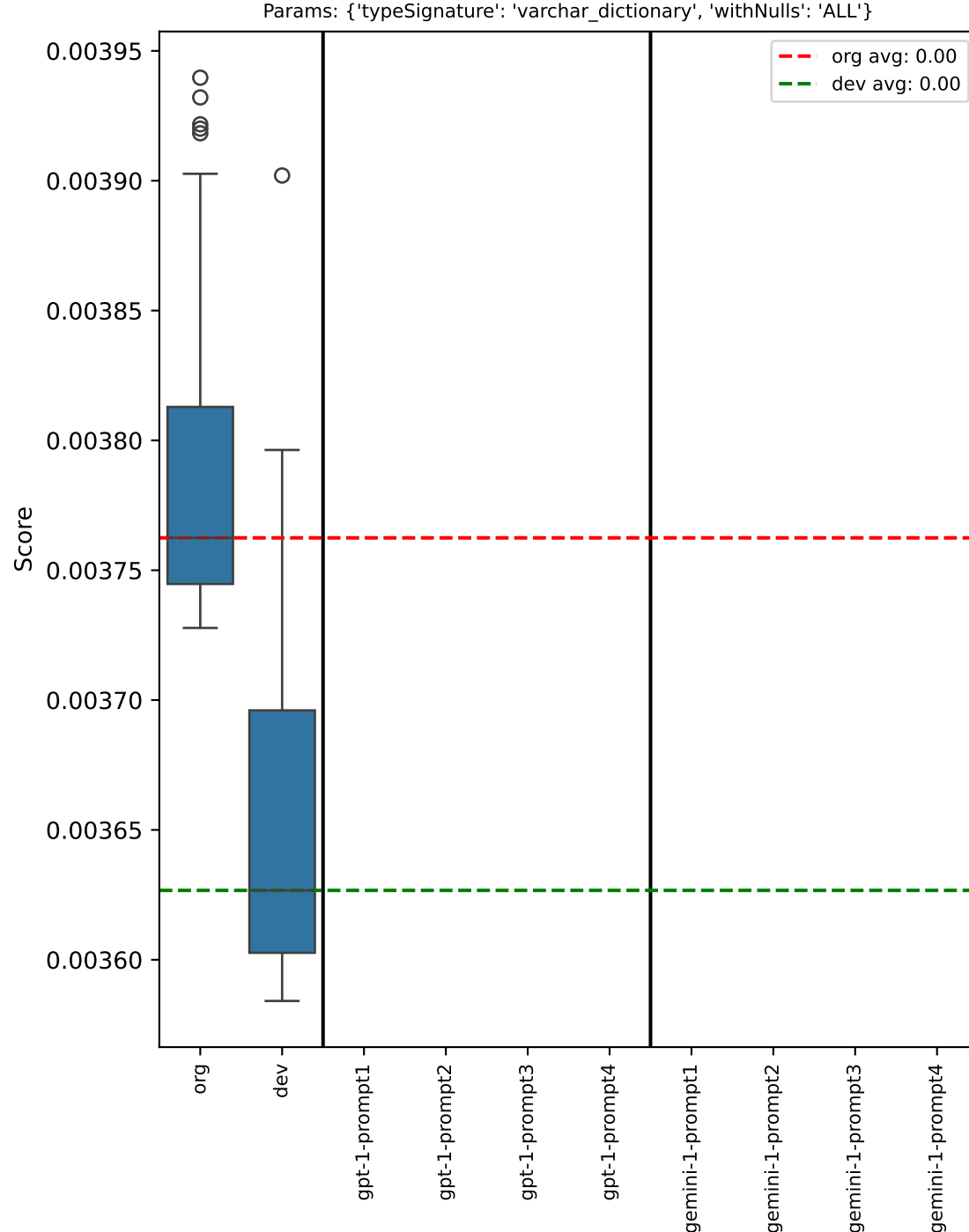
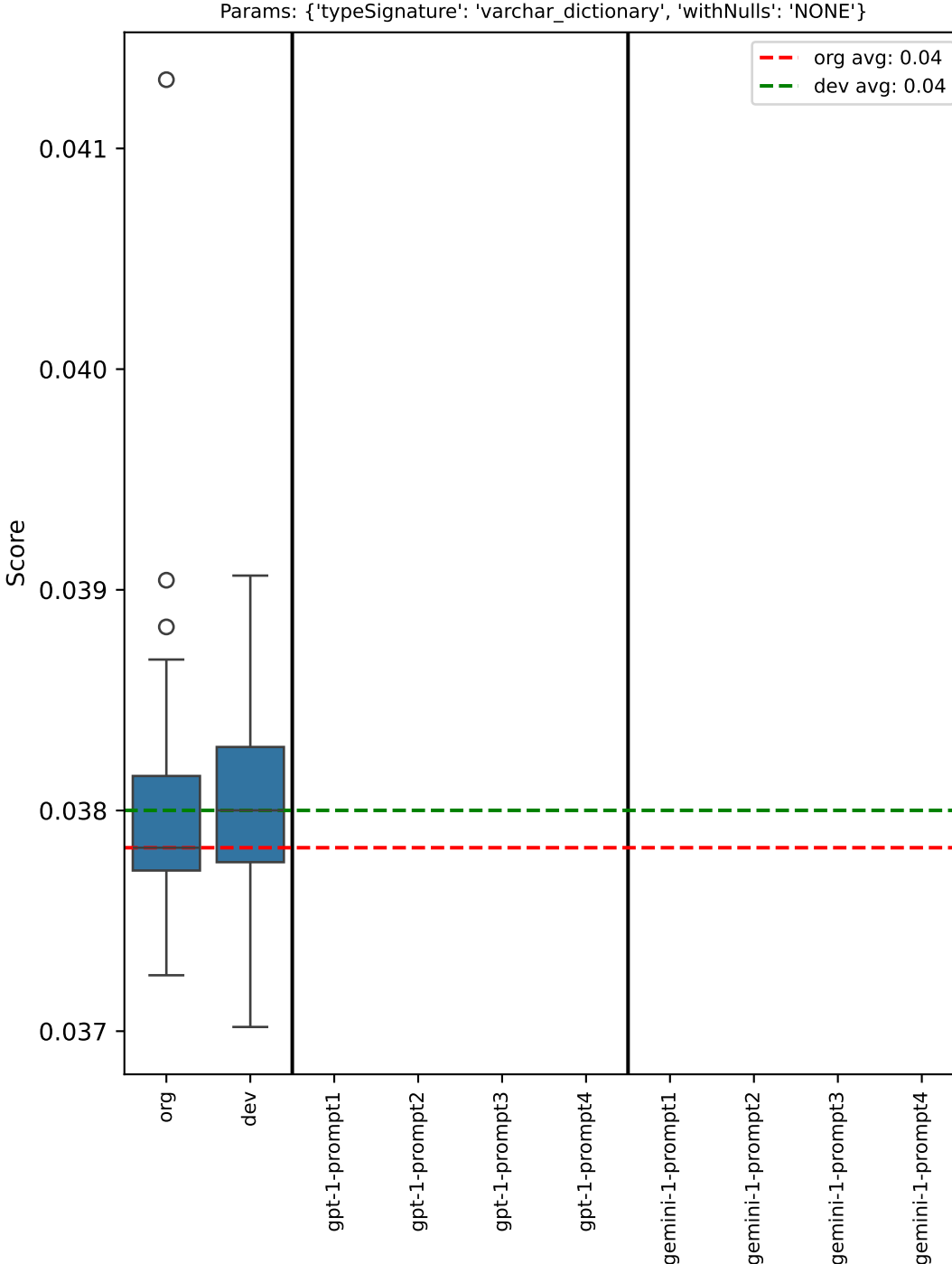
Params: {'typeSignature': 'varchar_direct', 'withNulls': 'ALL'}



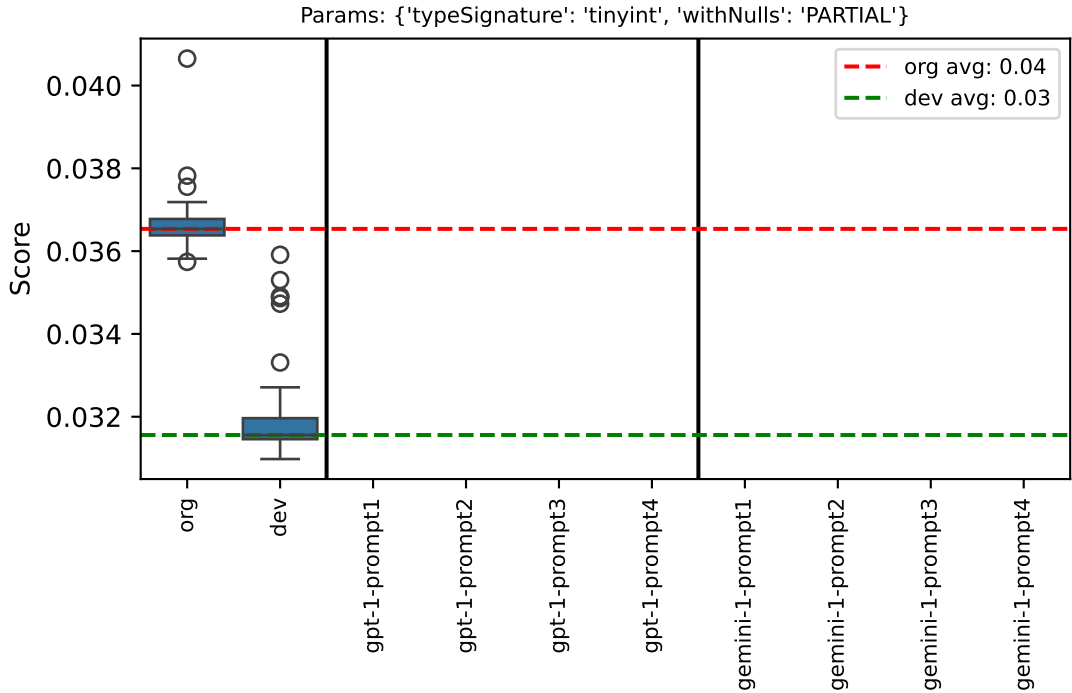
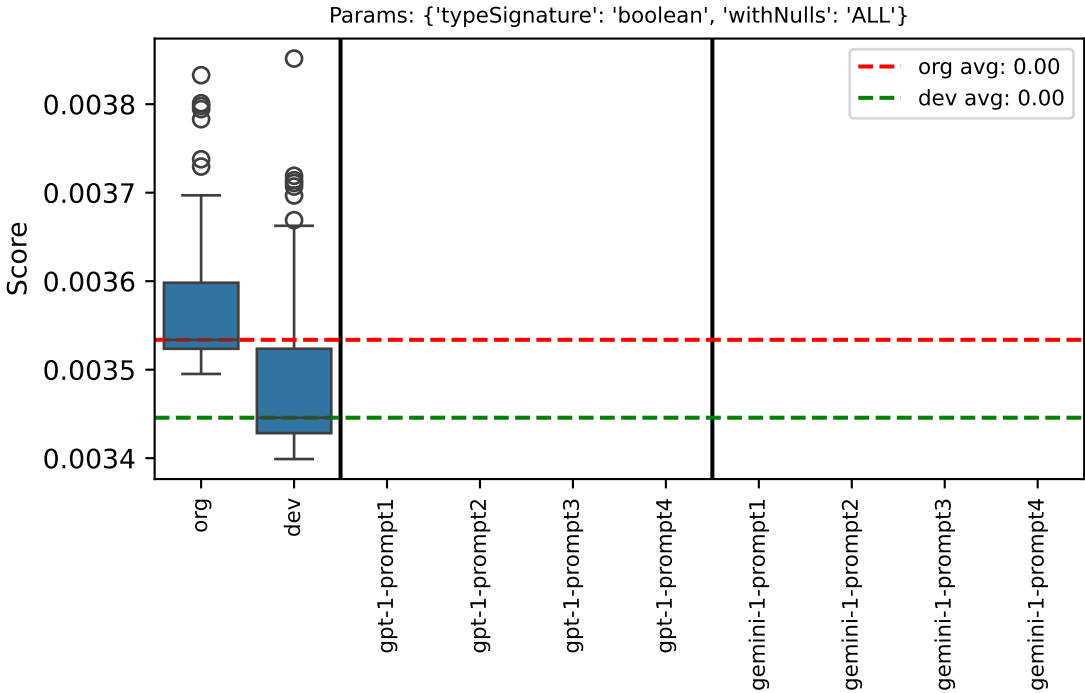
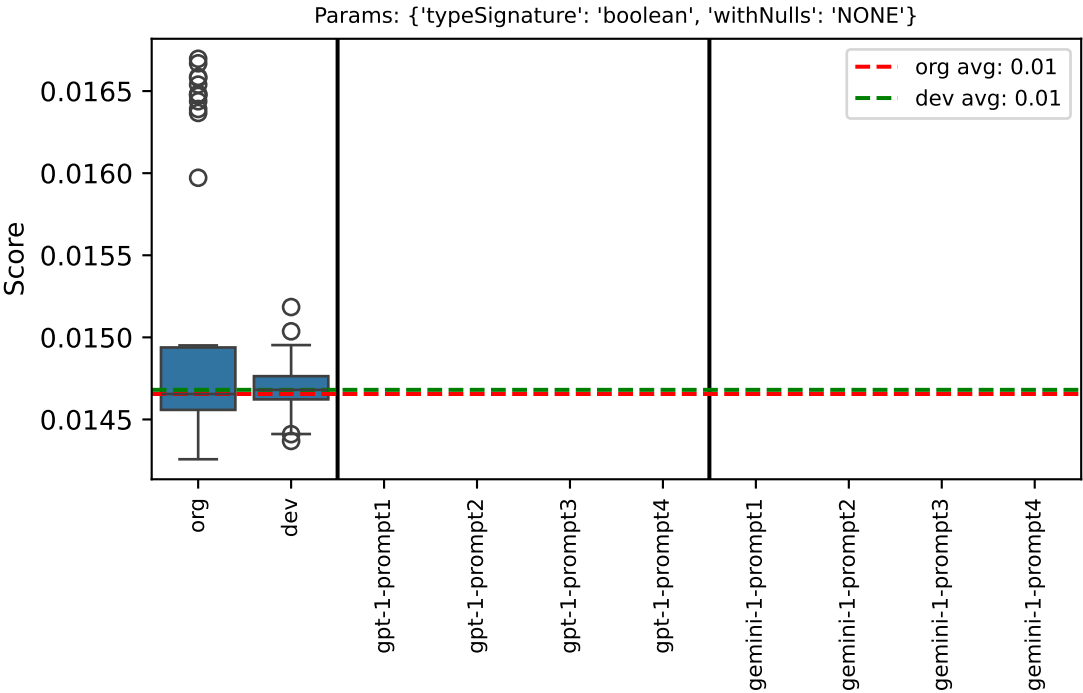
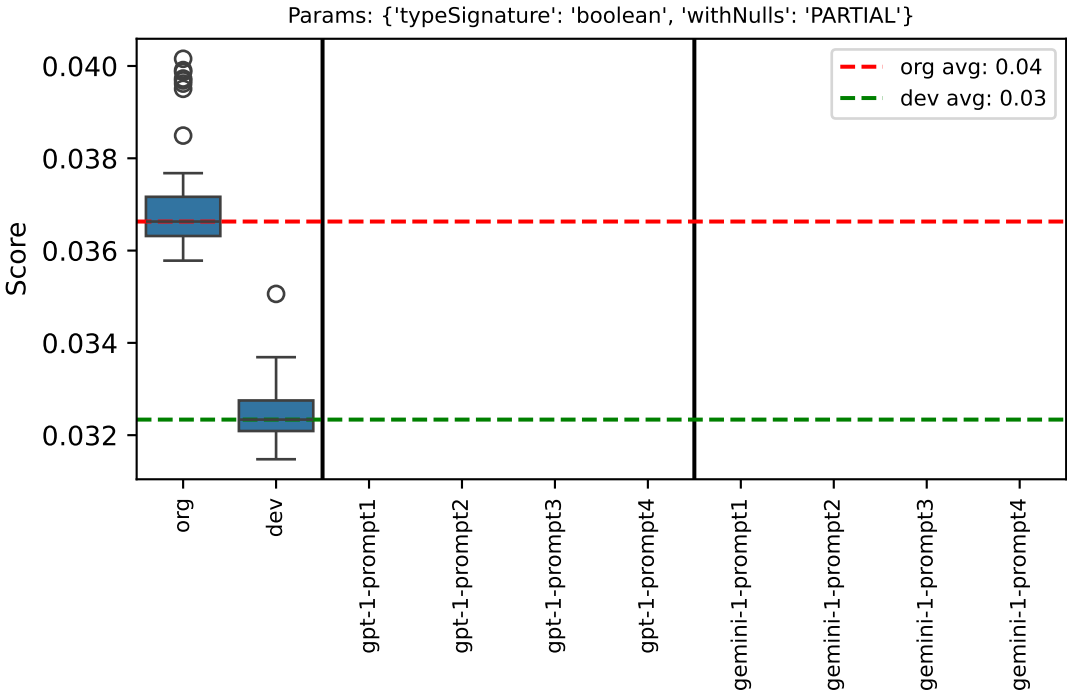
Params: {'typeSignature': 'varchar_dictionary', 'withNulls': 'PARTIAL'}



Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithIni

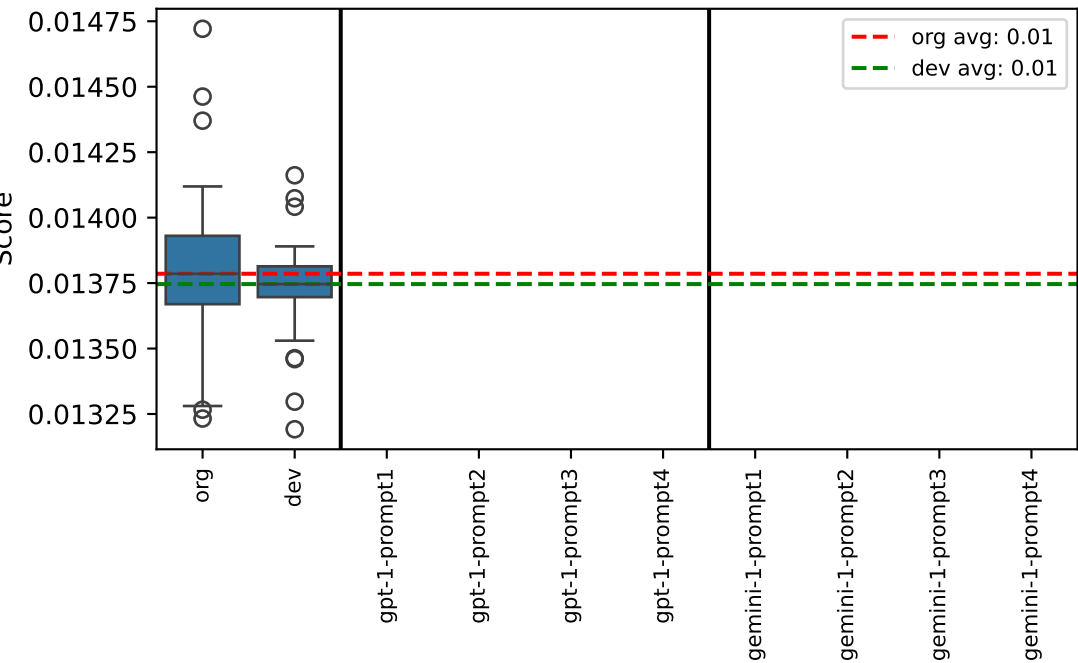


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni

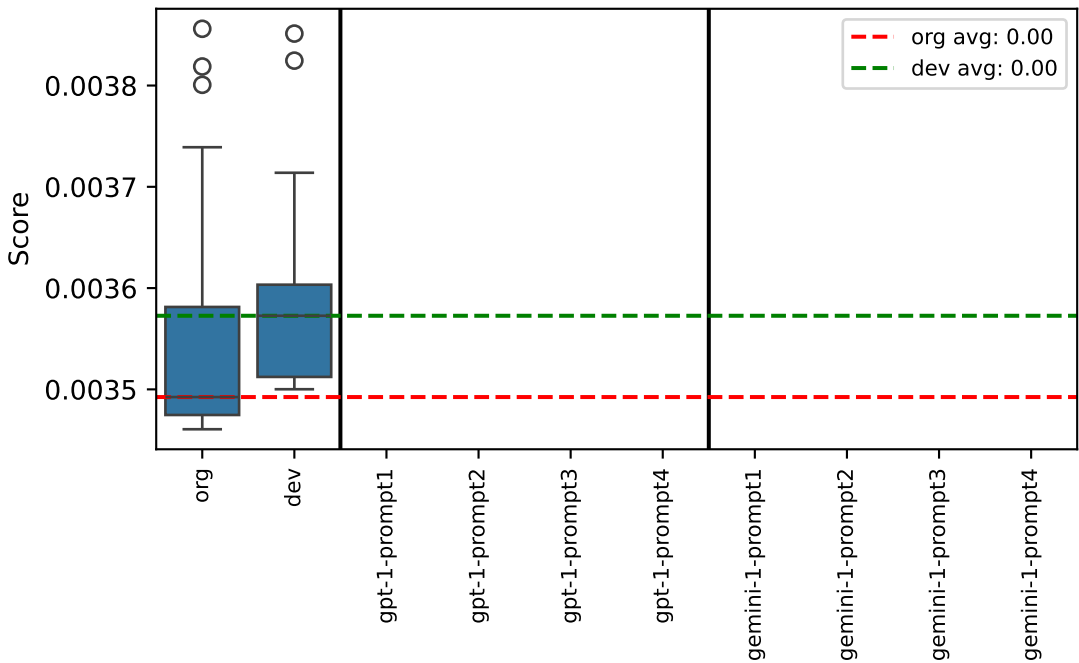


Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni

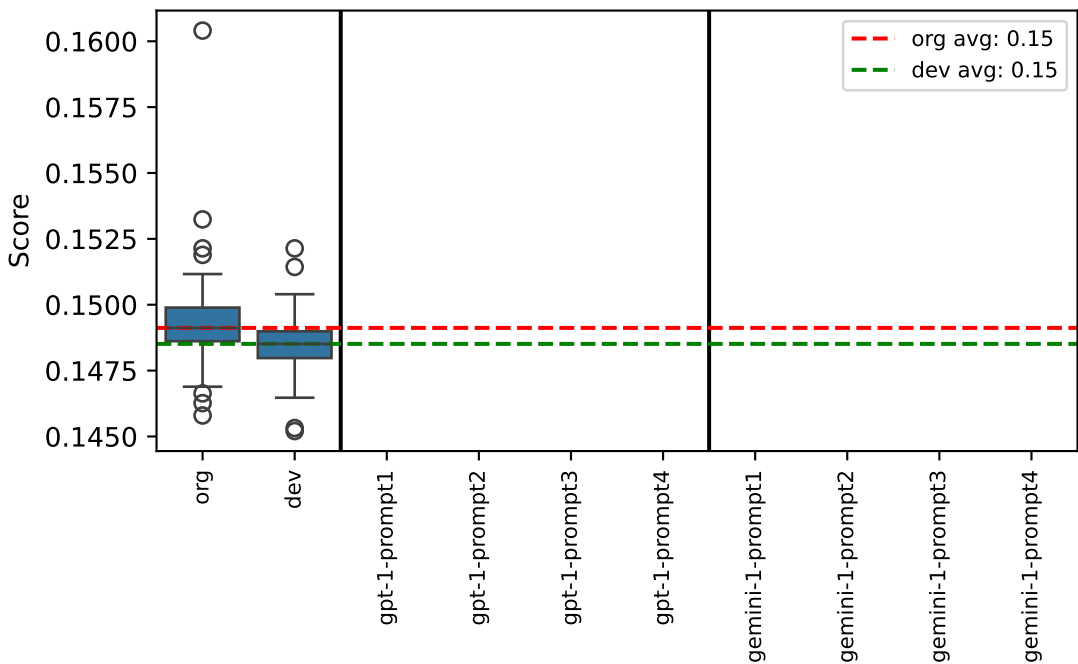
Params: {'typeSignature': 'tinyint', 'withNulls': 'NONE'}



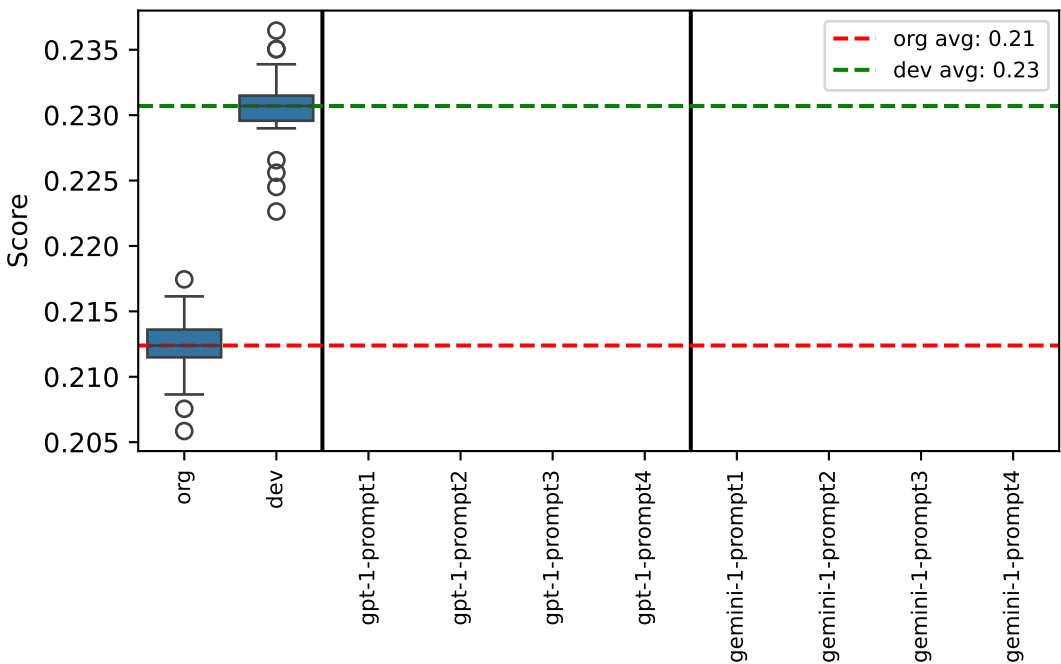
Params: {'typeSignature': 'tinyint', 'withNulls': 'ALL'}



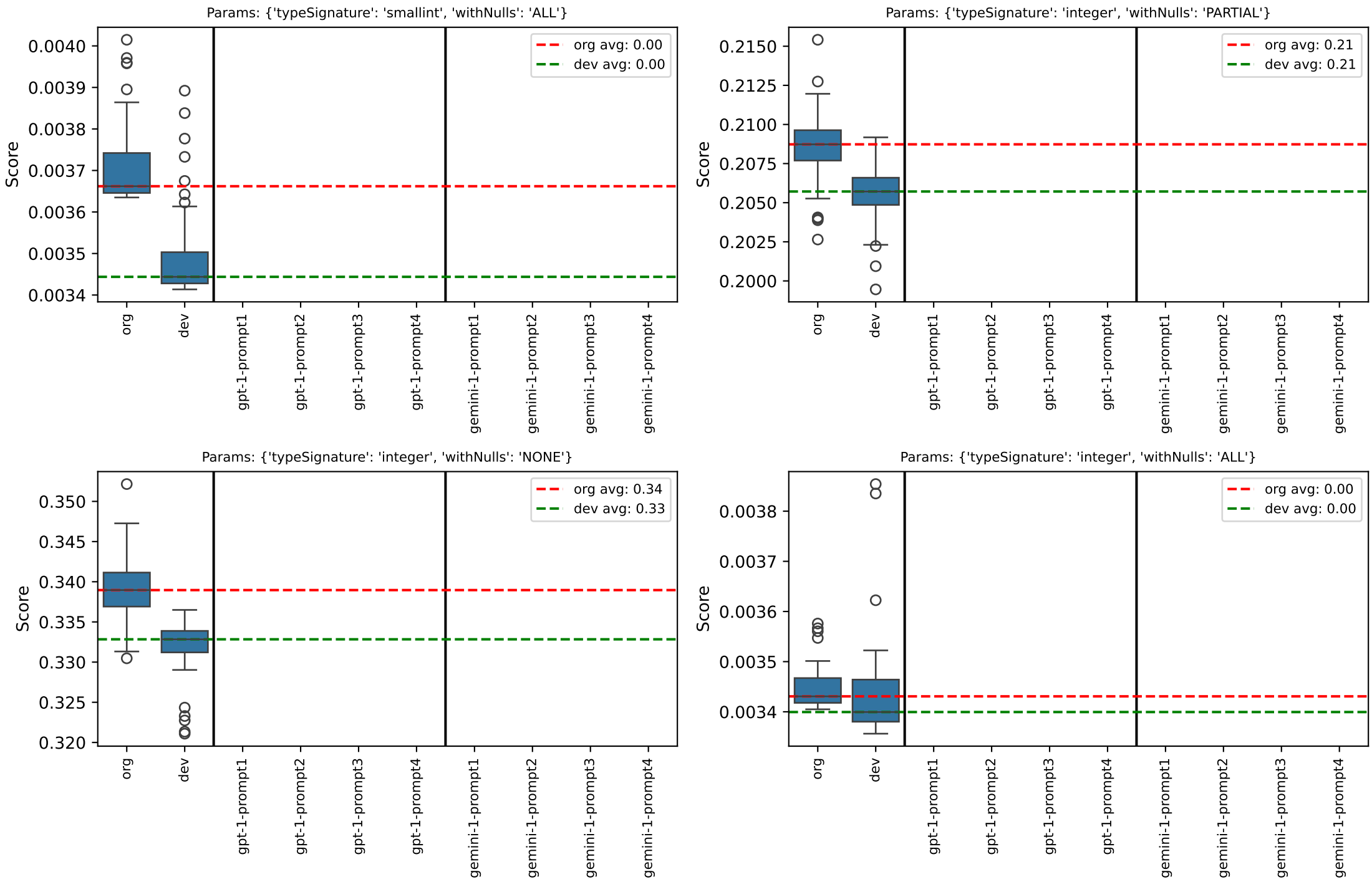
Params: {'typeSignature': 'smallint', 'withNulls': 'PARTIAL'}



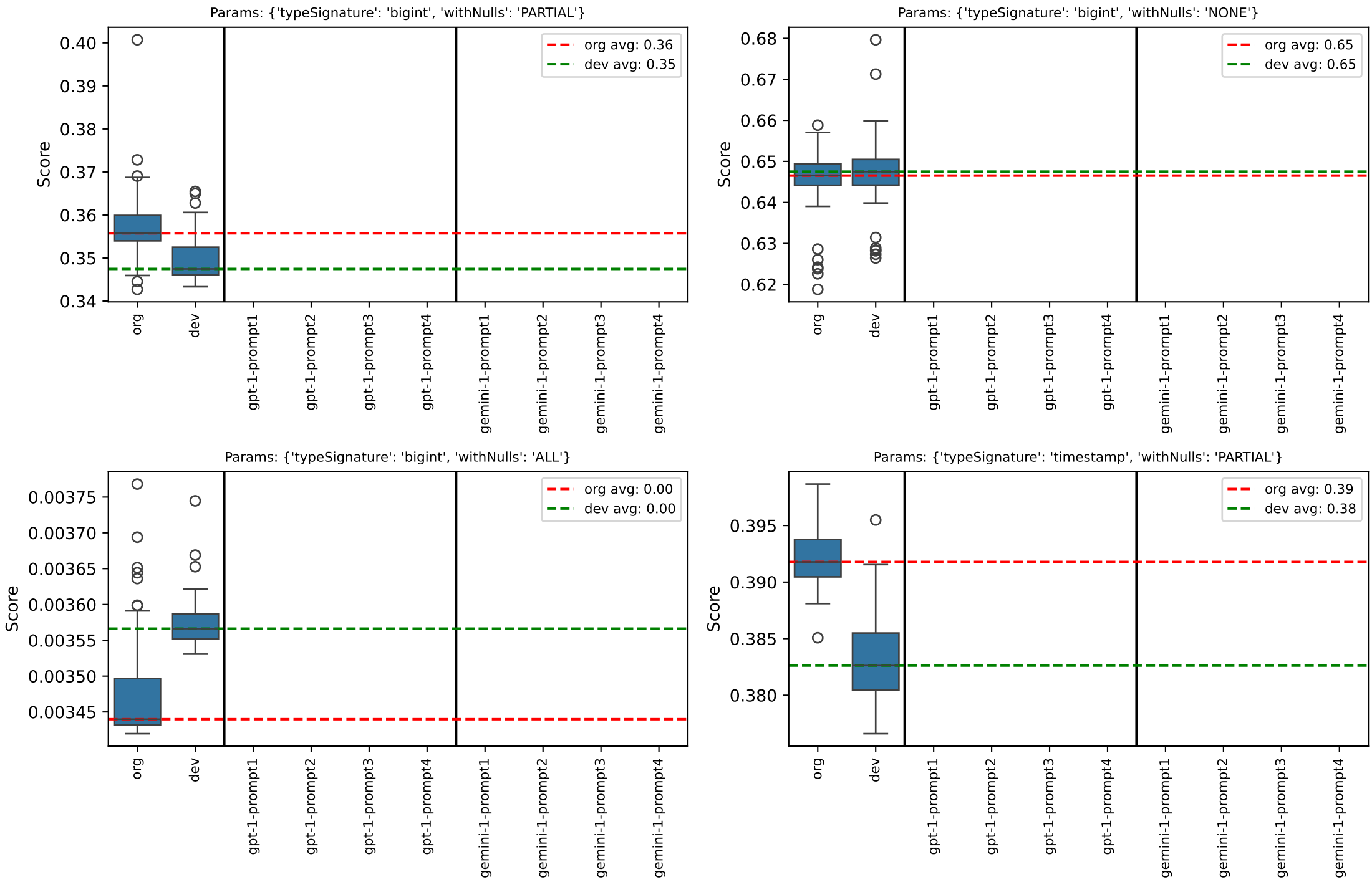
Params: {'typeSignature': 'smallint', 'withNulls': 'NONE'}



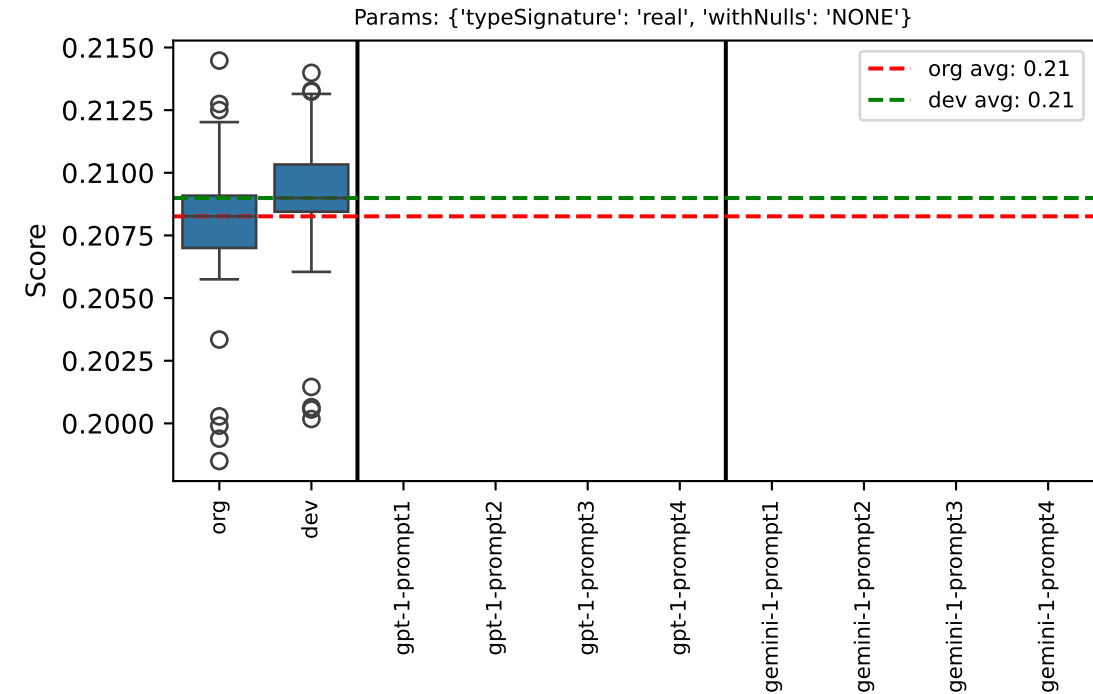
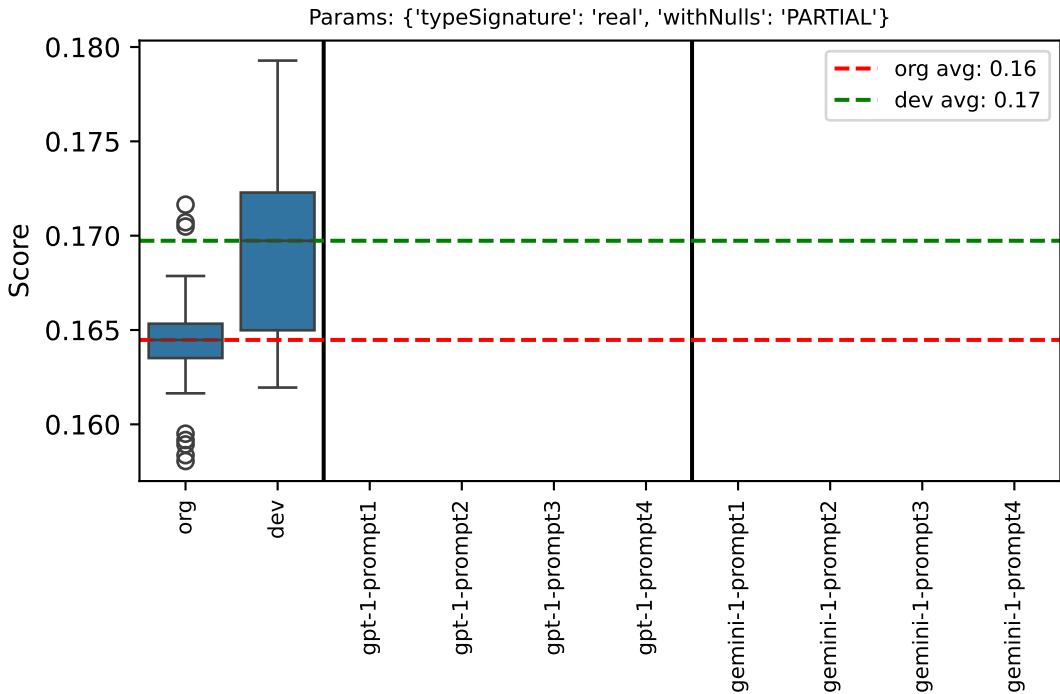
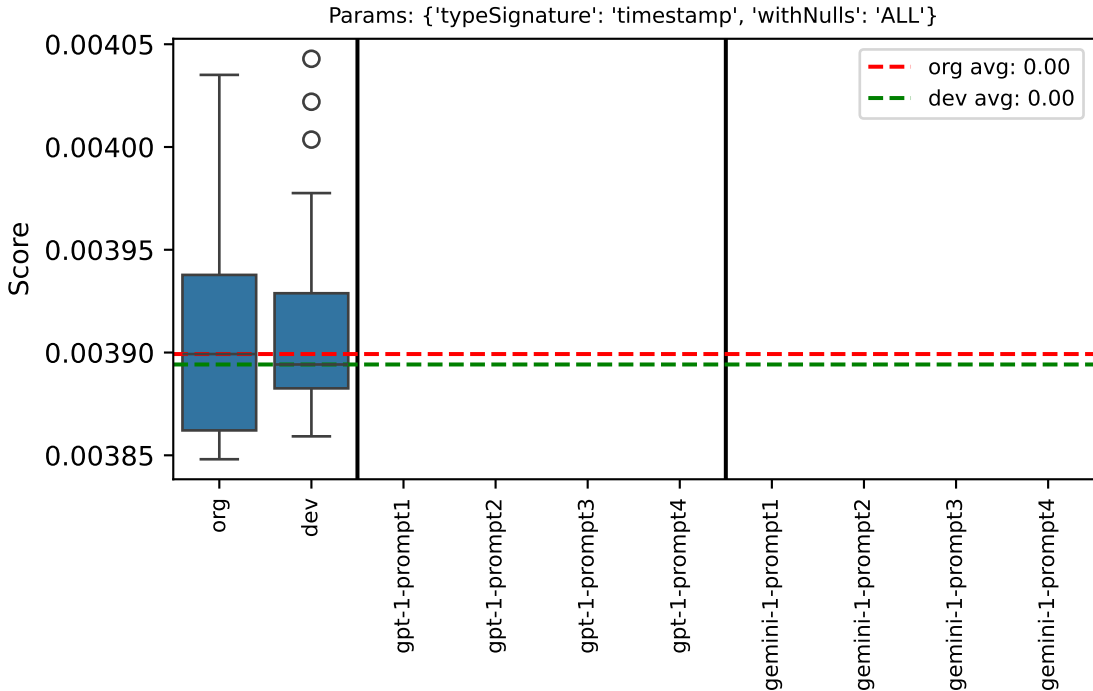
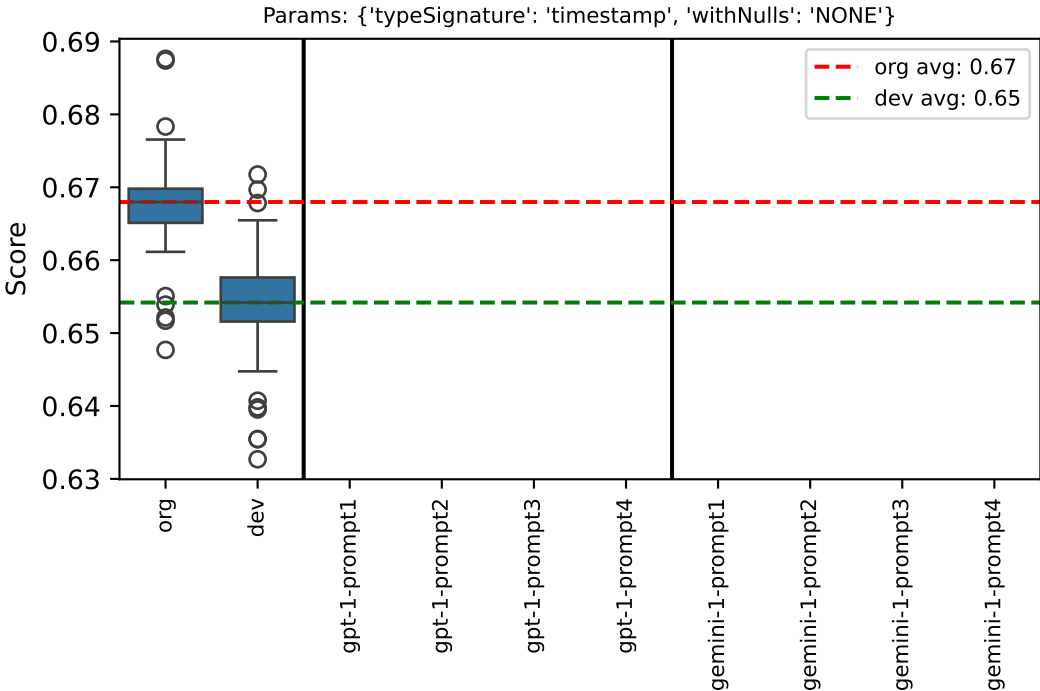
Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni



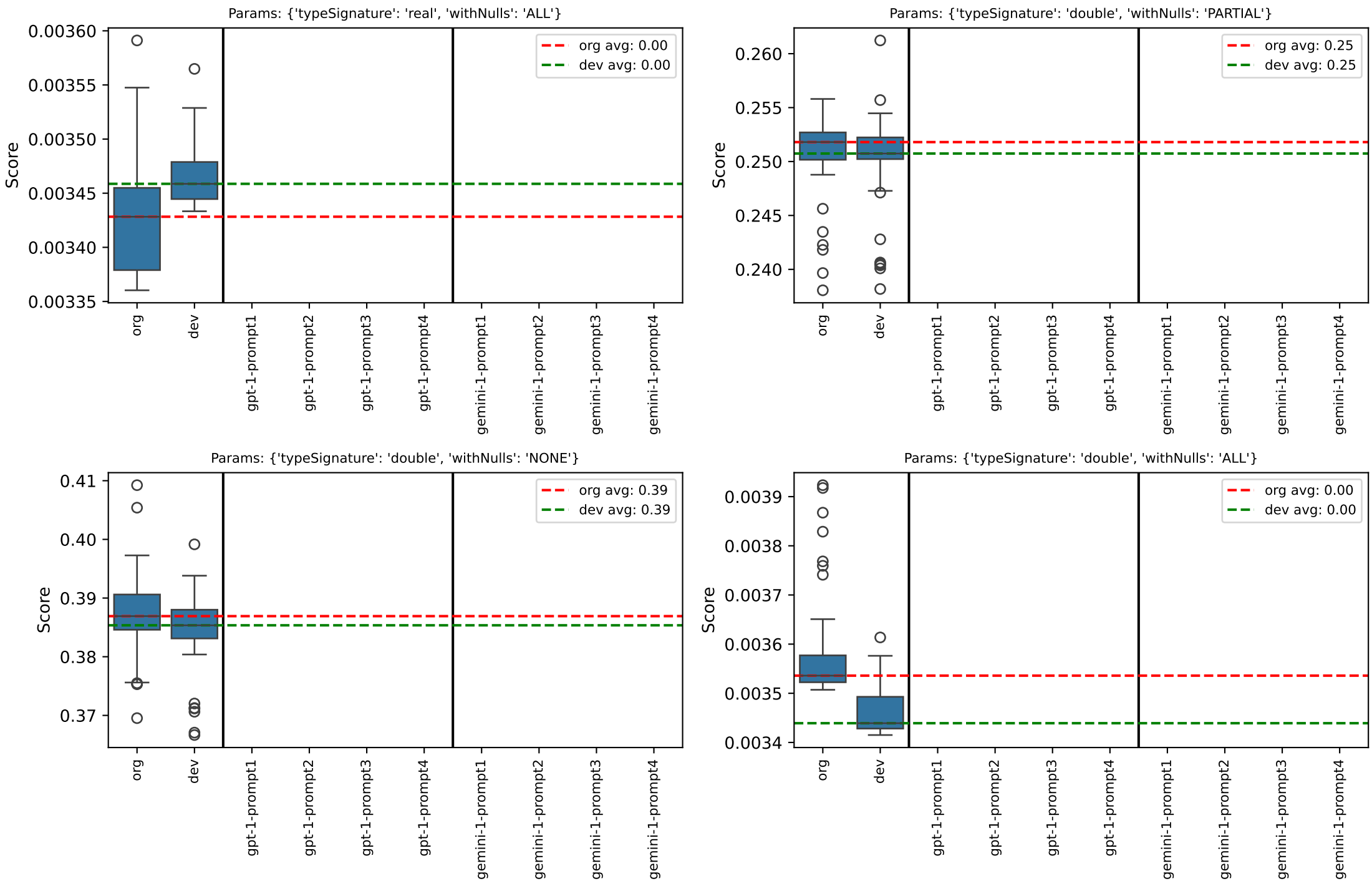
Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni



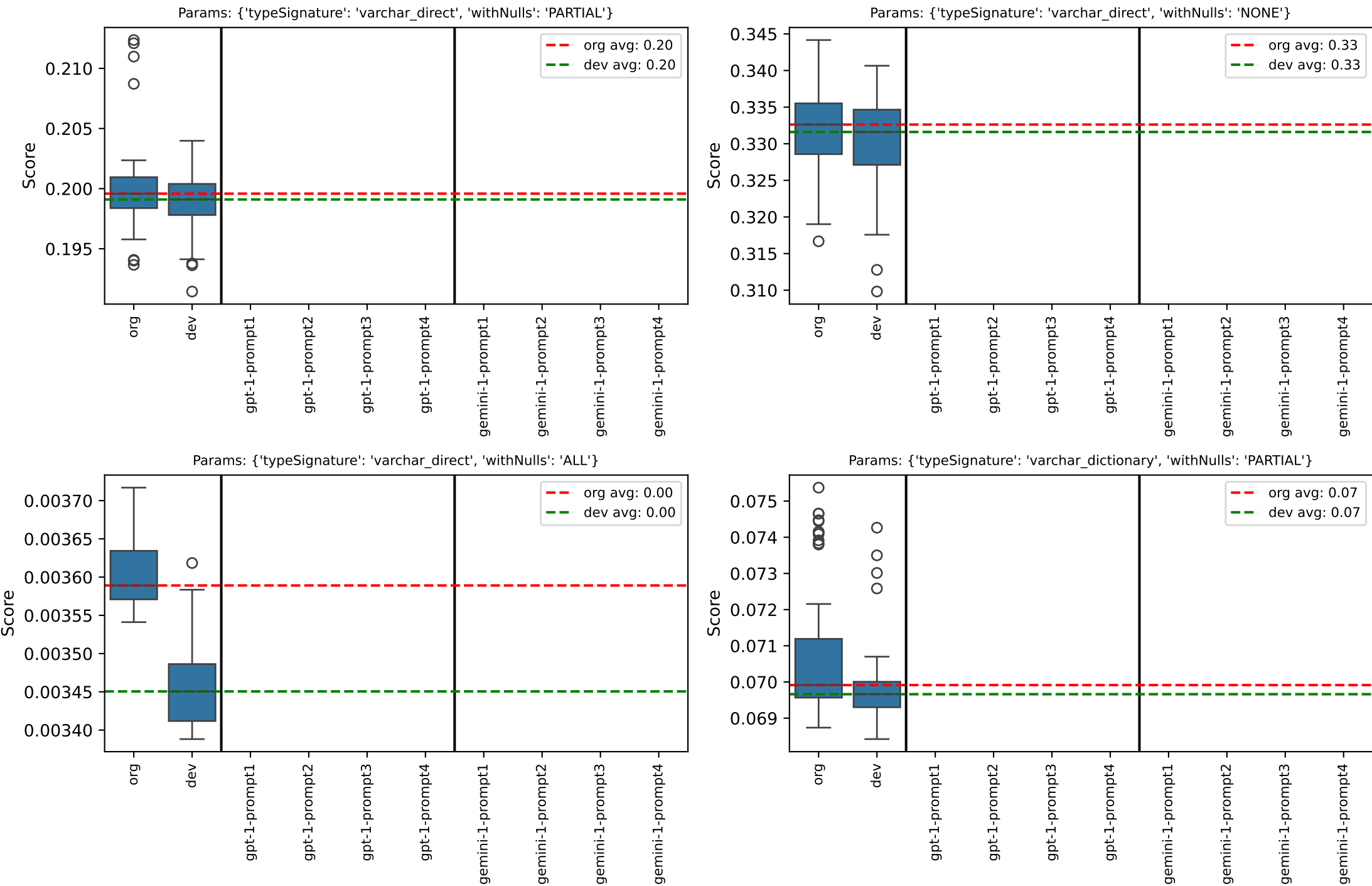
Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni



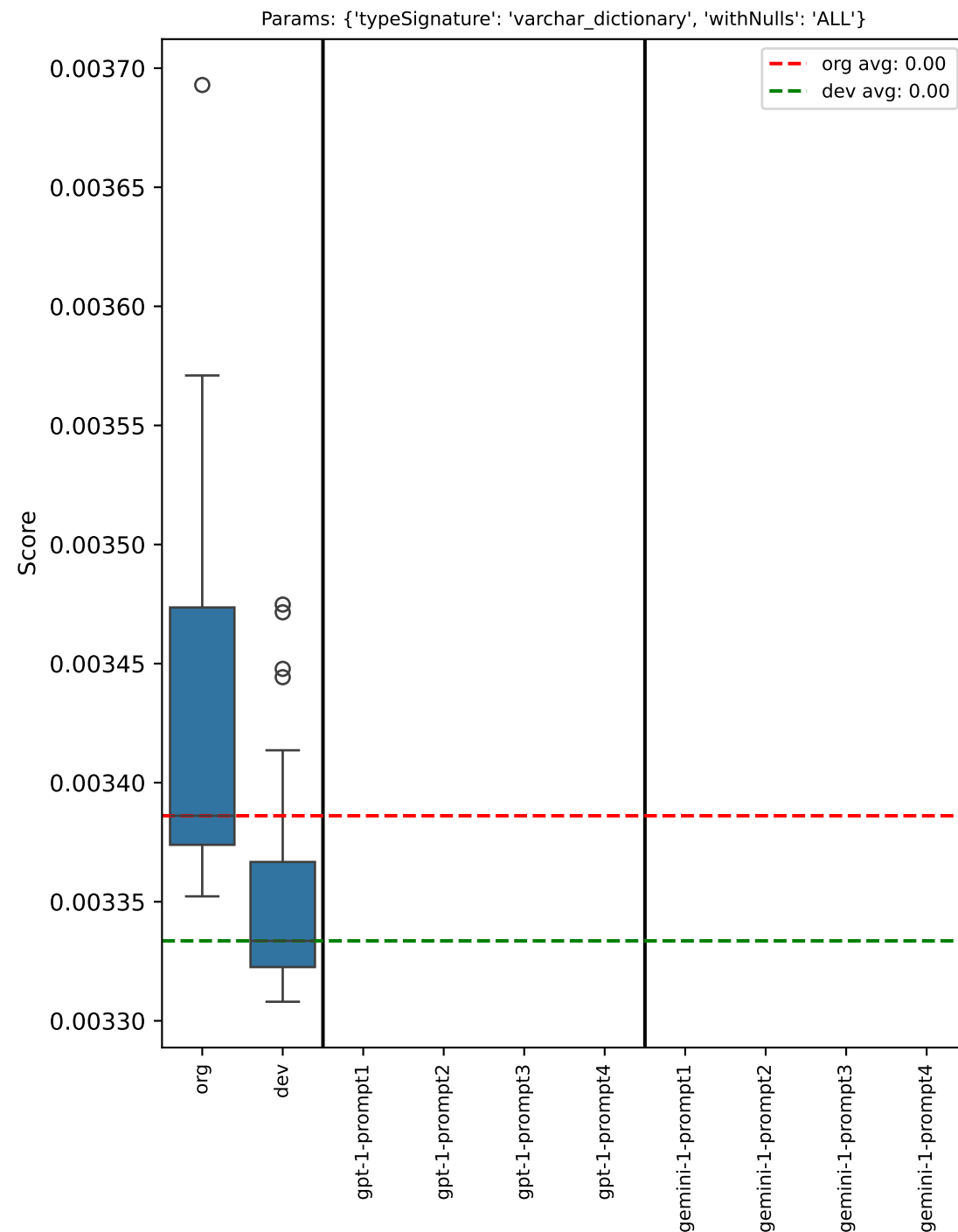
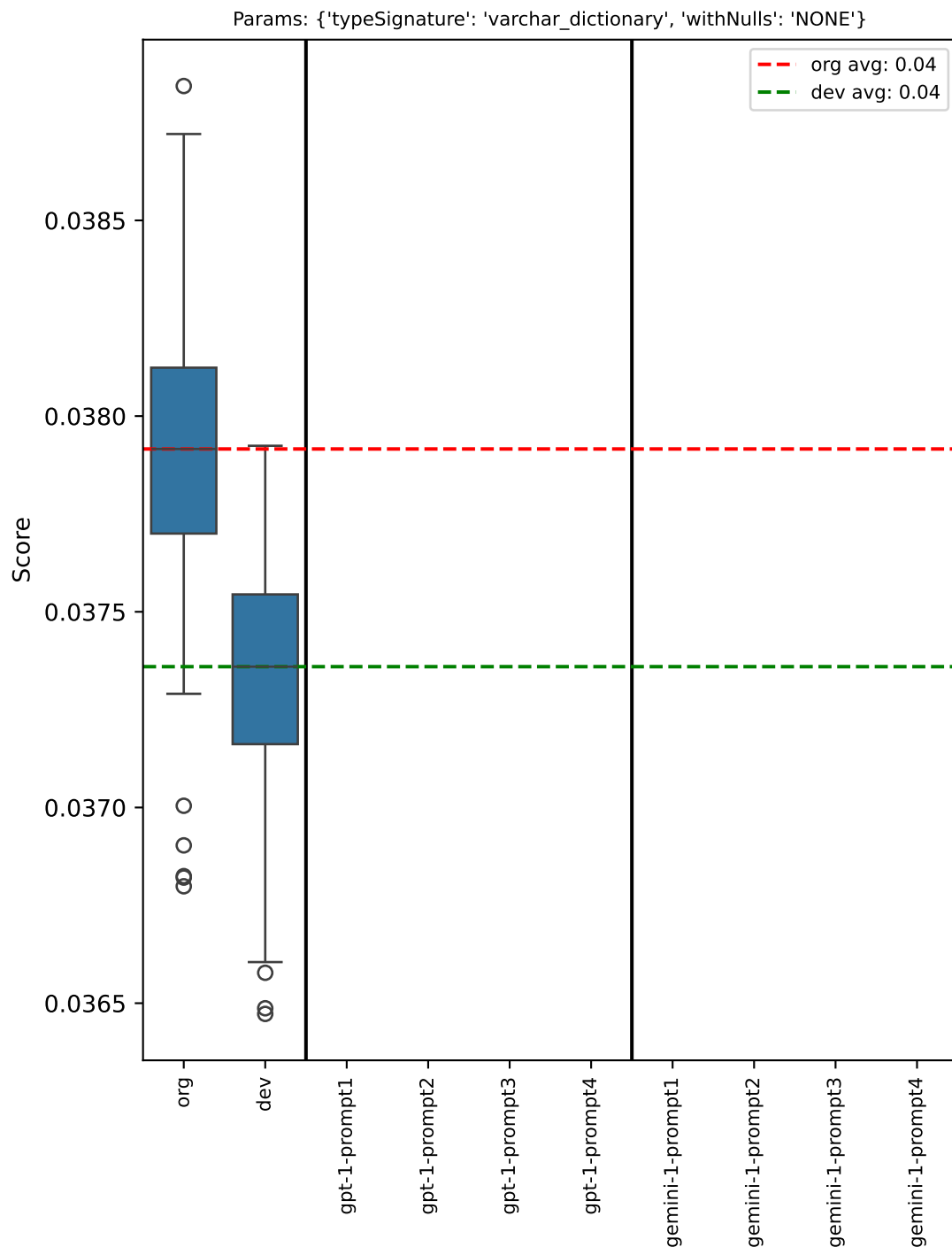
Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni



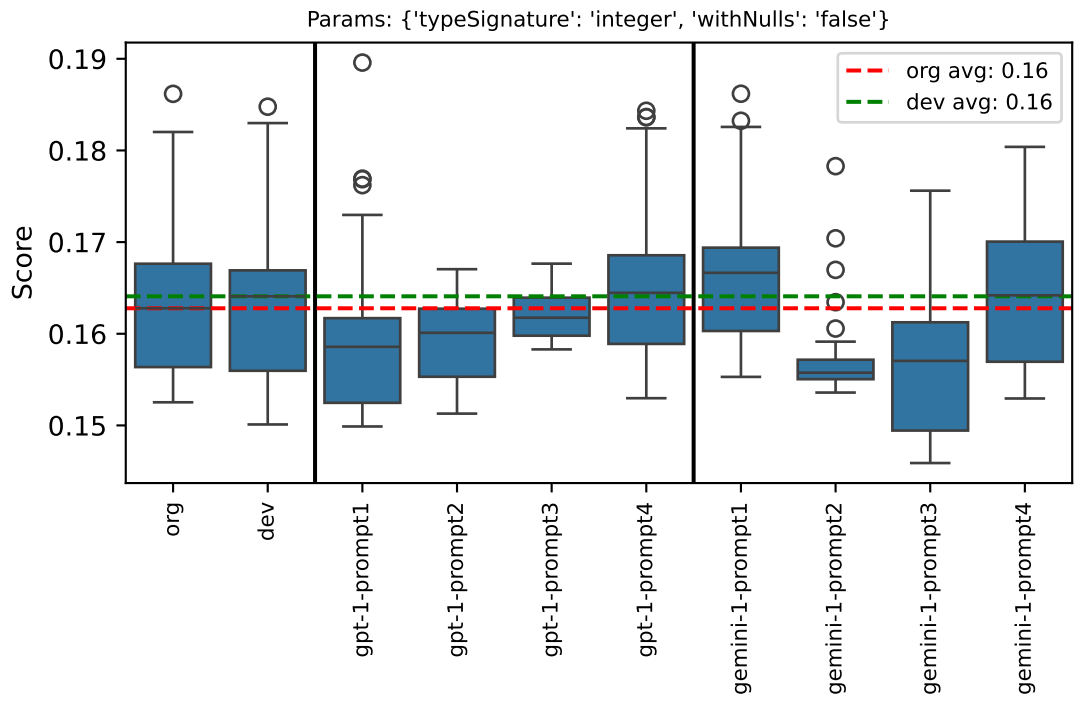
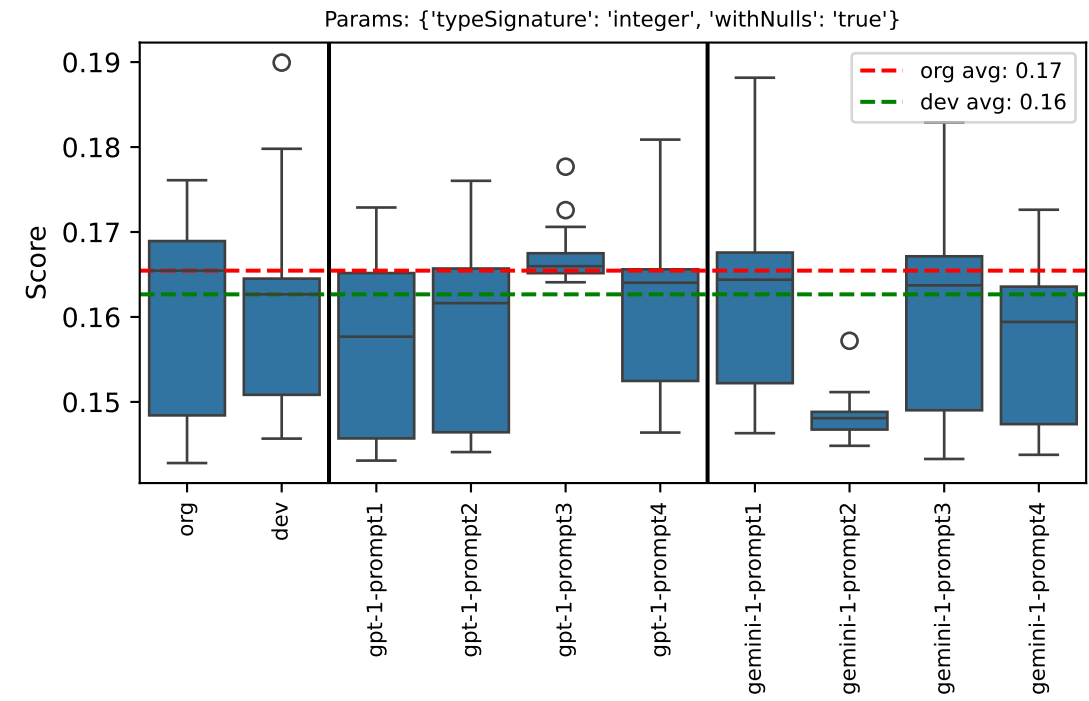
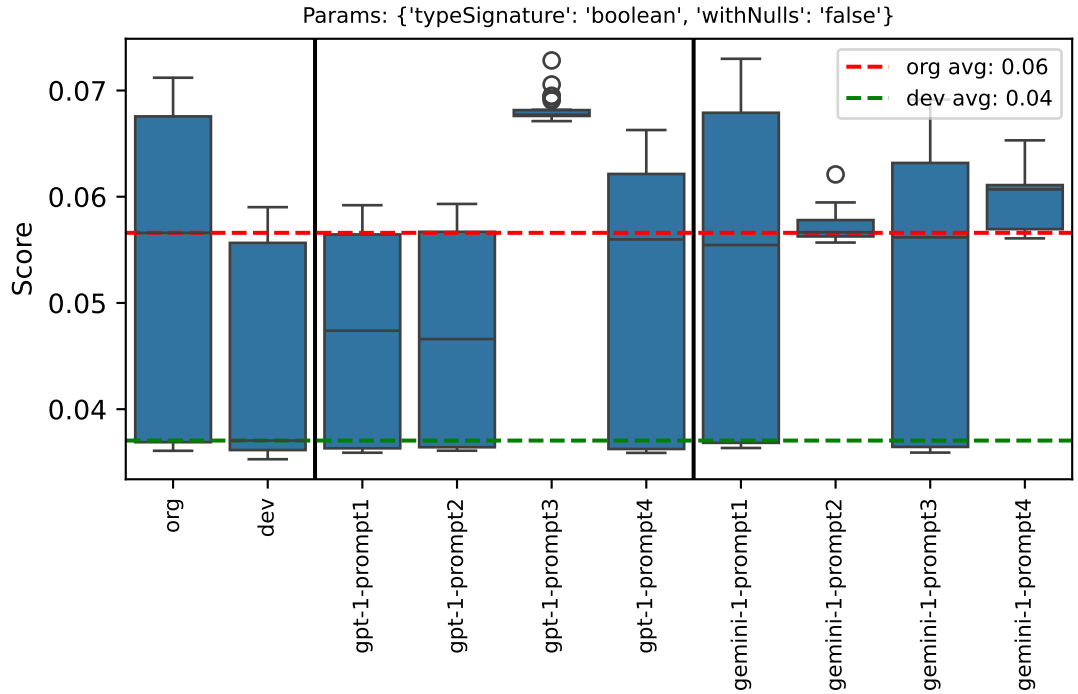
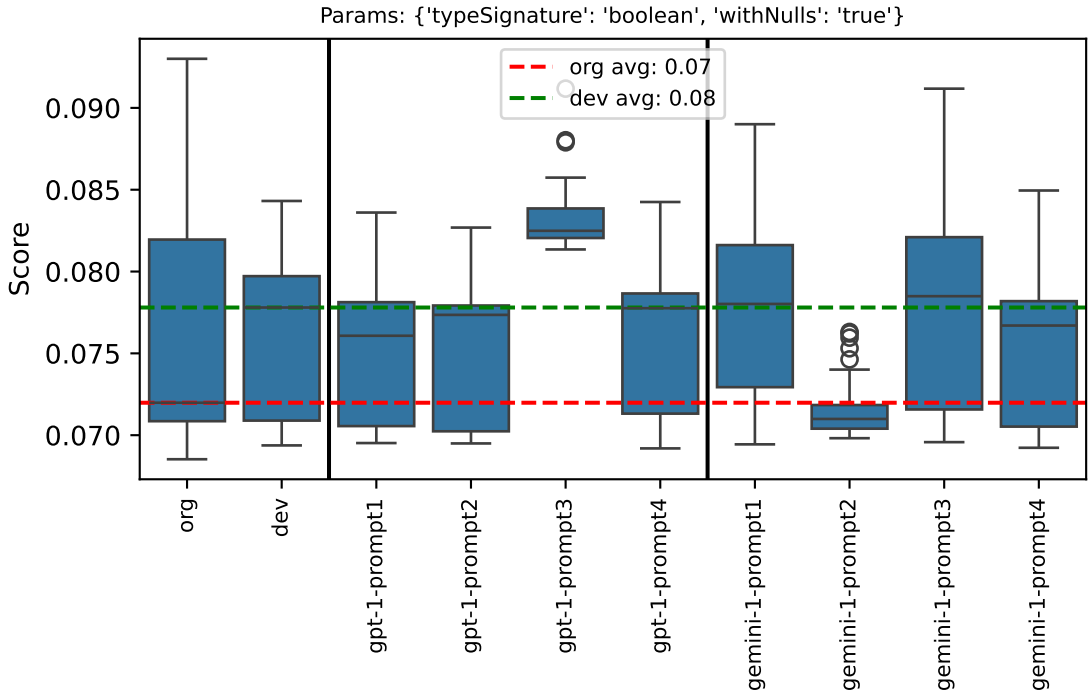
Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni



Benchmark: com.facebook.presto.orc.BenchmarkBatchStreamReadersWithZstd.readBlocksWithoutJni

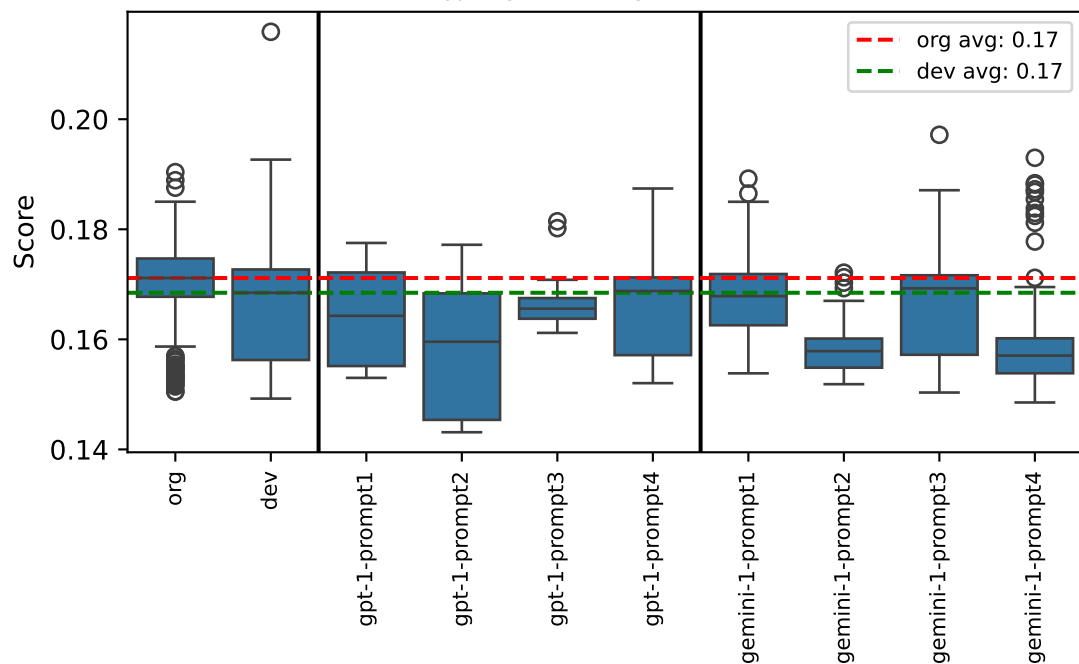


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

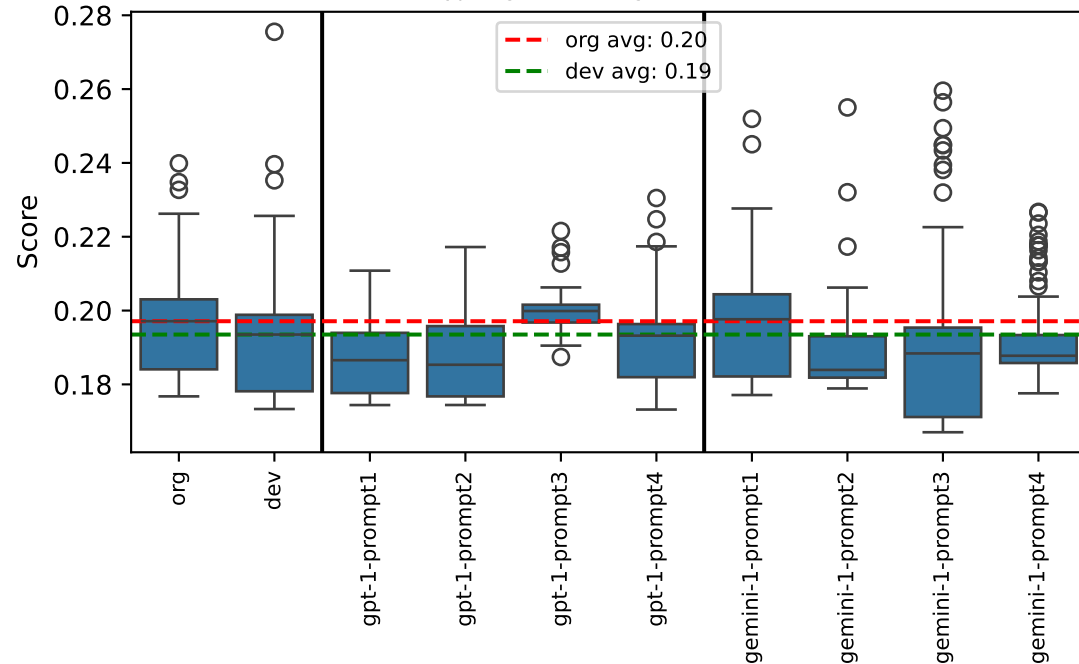


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

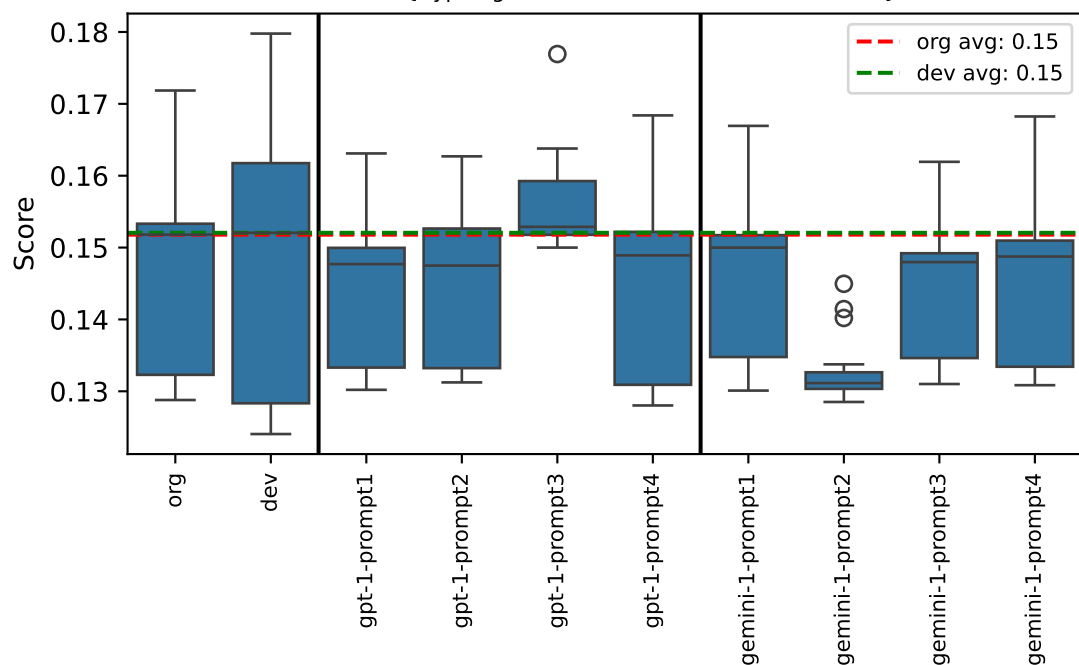
Params: {'typeSignature': 'bigint', 'withNulls': 'true'}



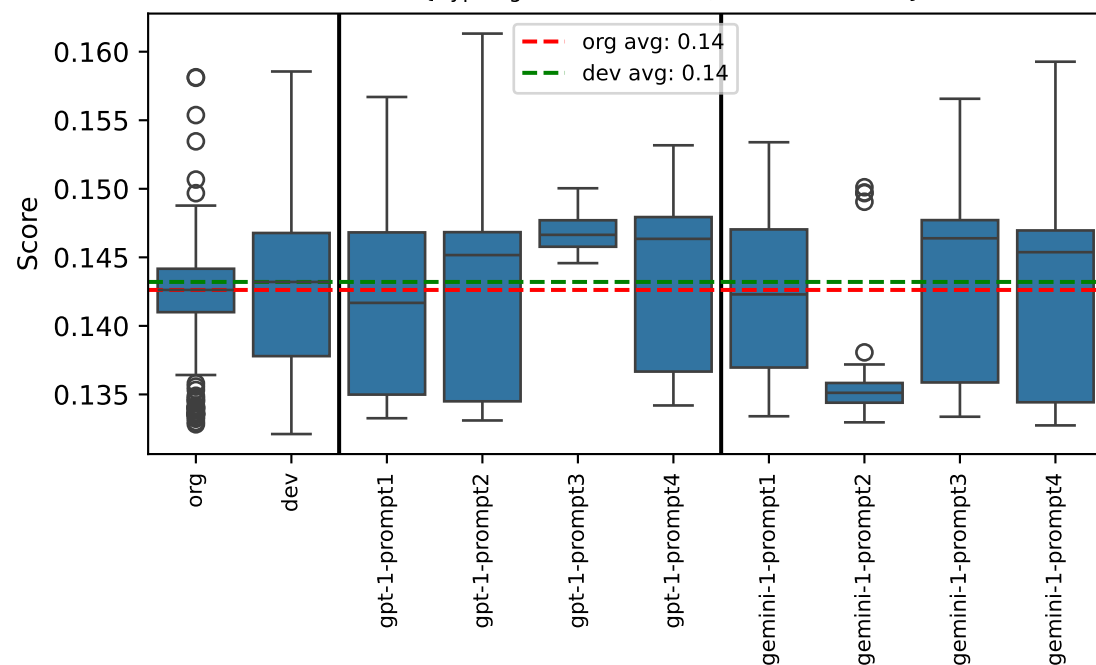
Params: {'typeSignature': 'bigint', 'withNulls': 'false'}



Params: {'typeSignature': 'smallint', 'withNulls': 'true'}

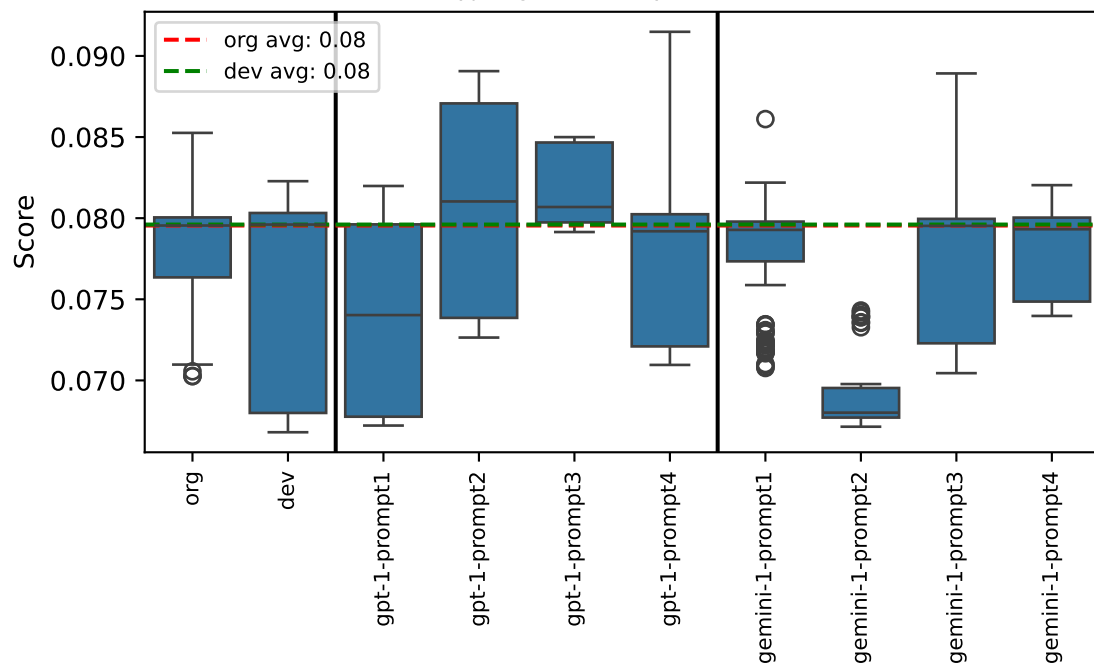


Params: {'typeSignature': 'smallint', 'withNulls': 'false'}

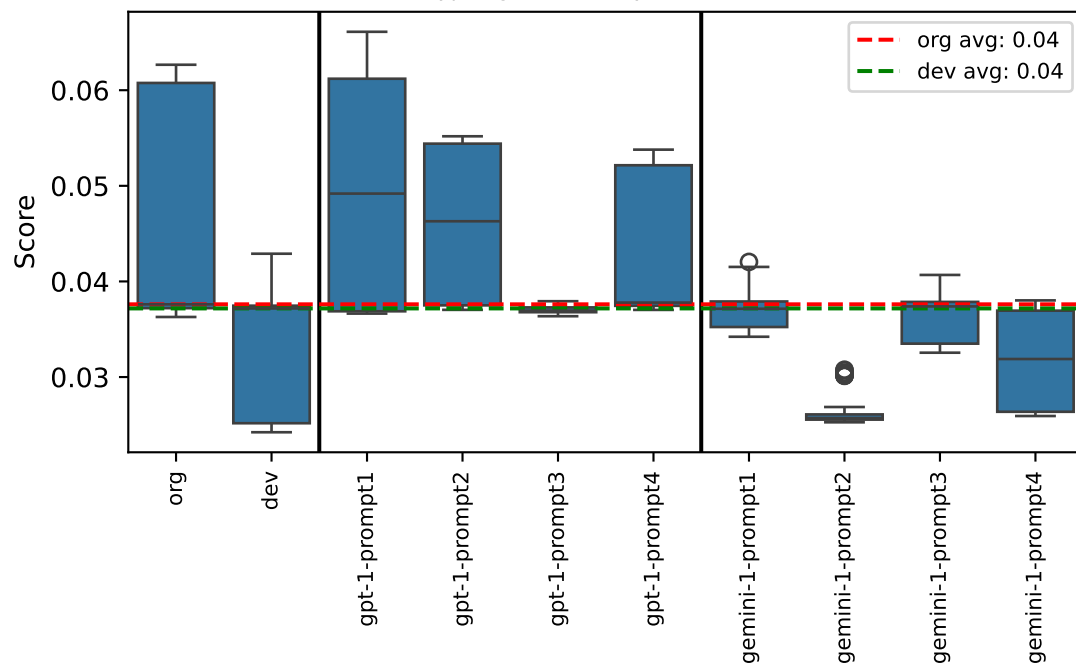


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

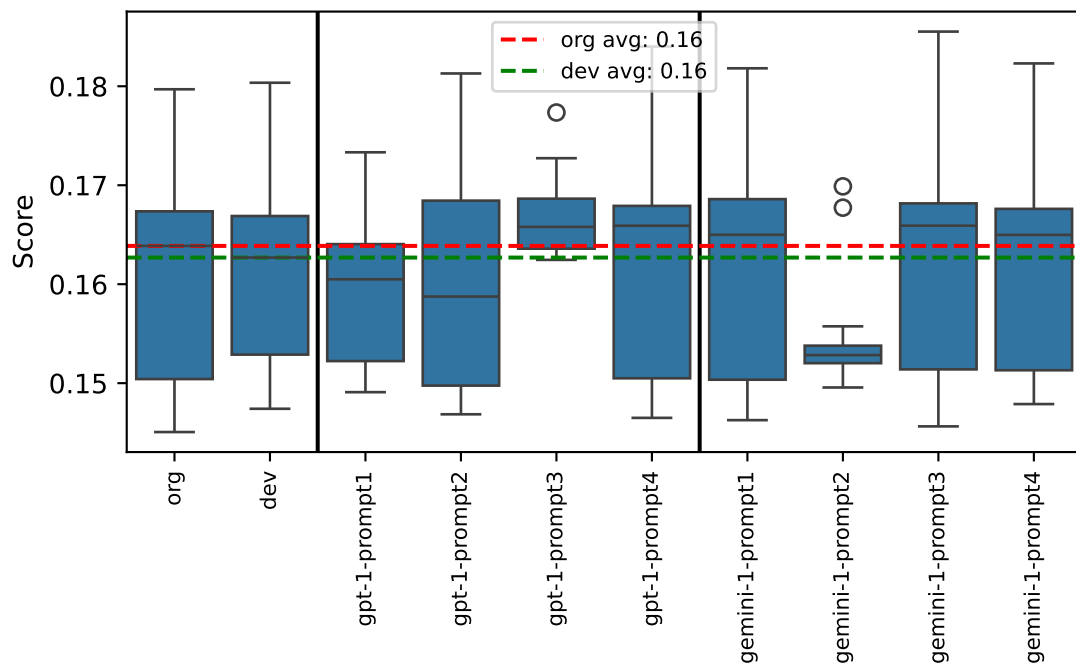
Params: {'typeSignature': 'tinyint', 'withNulls': 'true'}



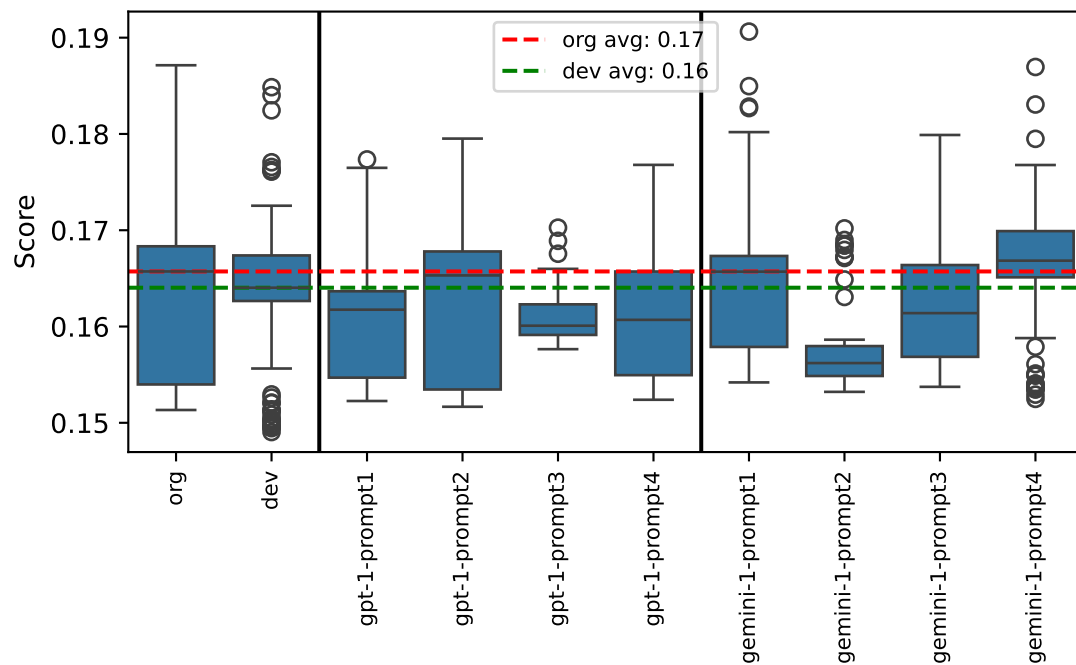
Params: {'typeSignature': 'tinyint', 'withNulls': 'false'}



Params: {'typeSignature': 'date', 'withNulls': 'true'}

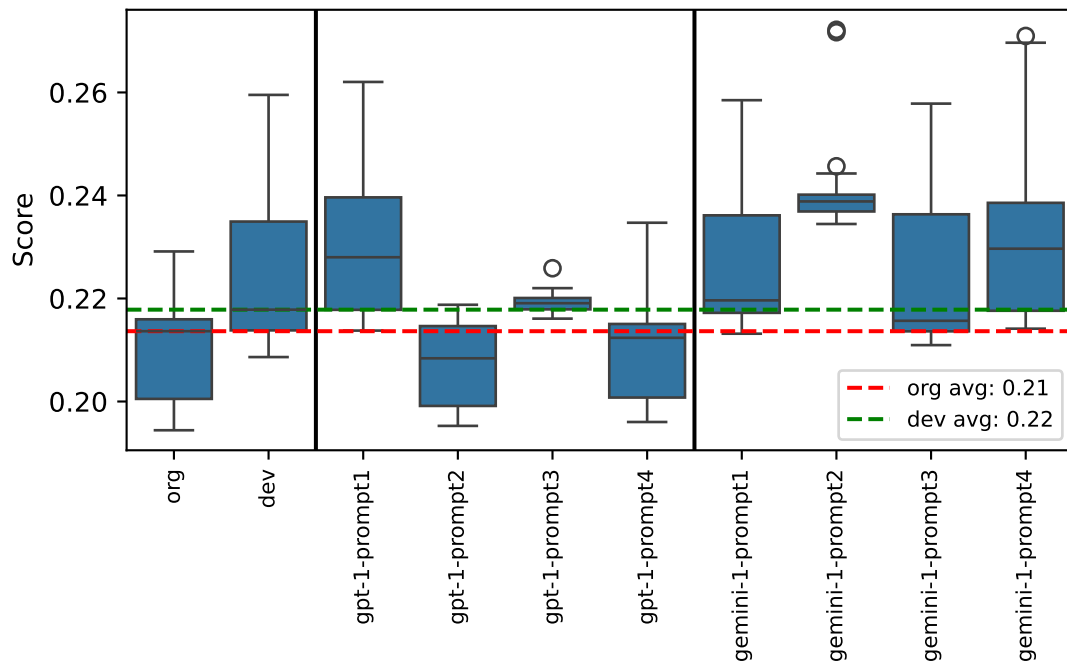


Params: {'typeSignature': 'date', 'withNulls': 'false'}

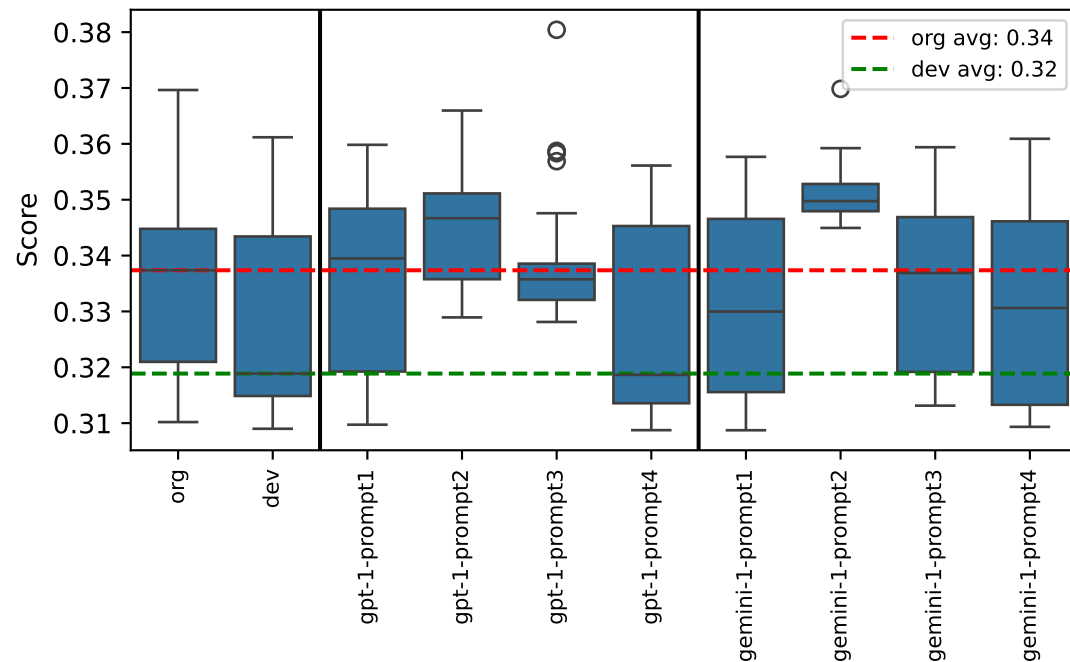


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

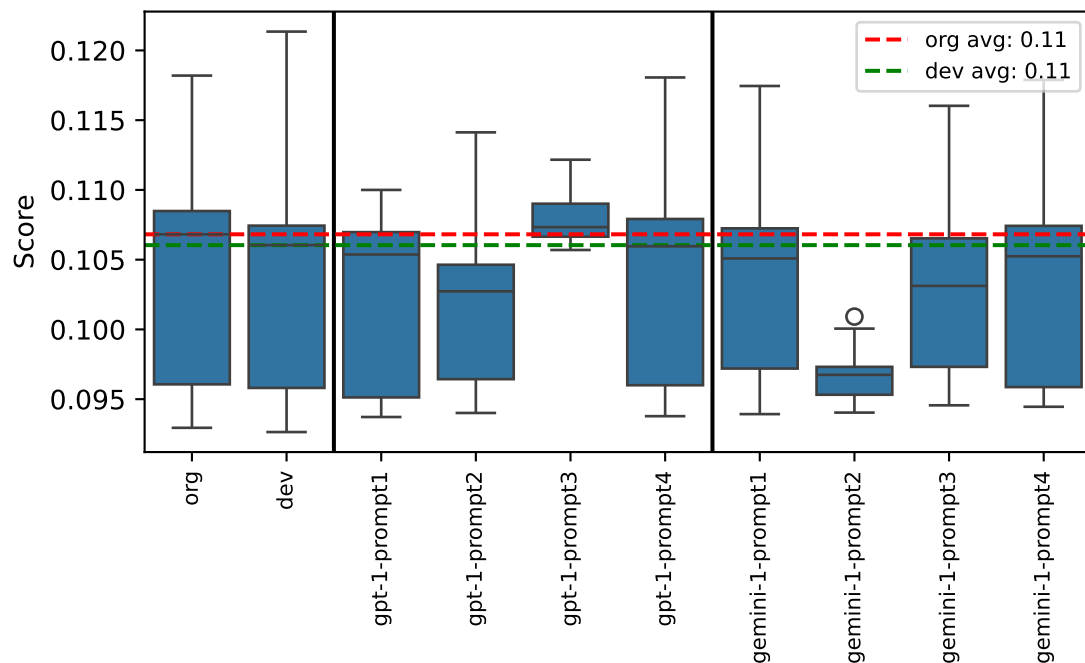
Params: {'typeSignature': 'timestamp', 'withNulls': 'true'}



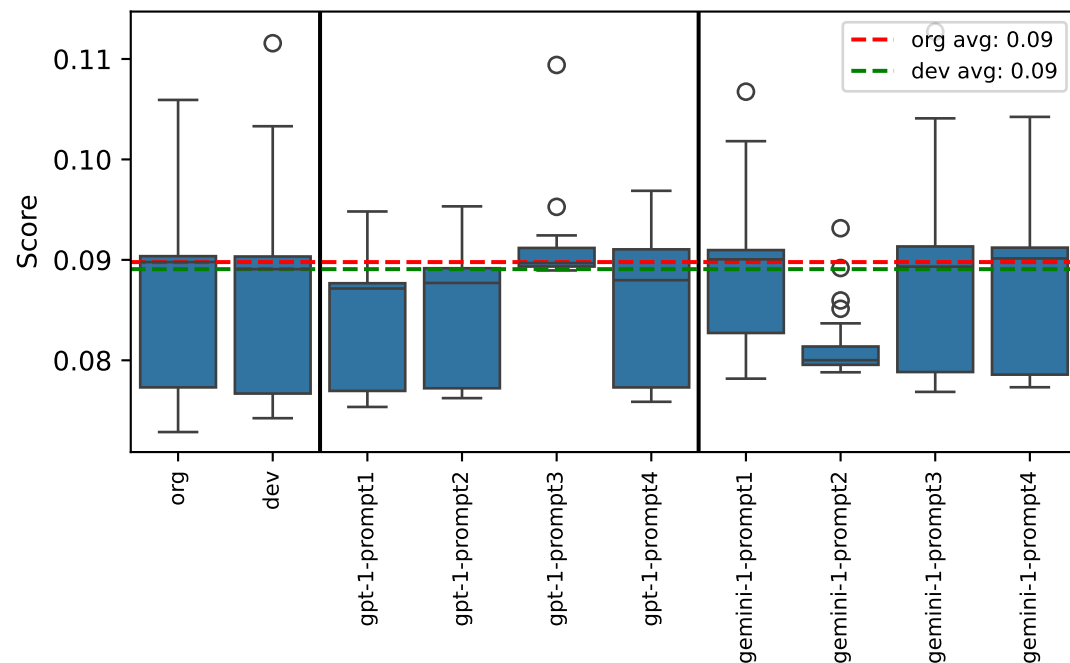
Params: {'typeSignature': 'timestamp', 'withNulls': 'false'}



Params: {'typeSignature': 'real', 'withNulls': 'true'}

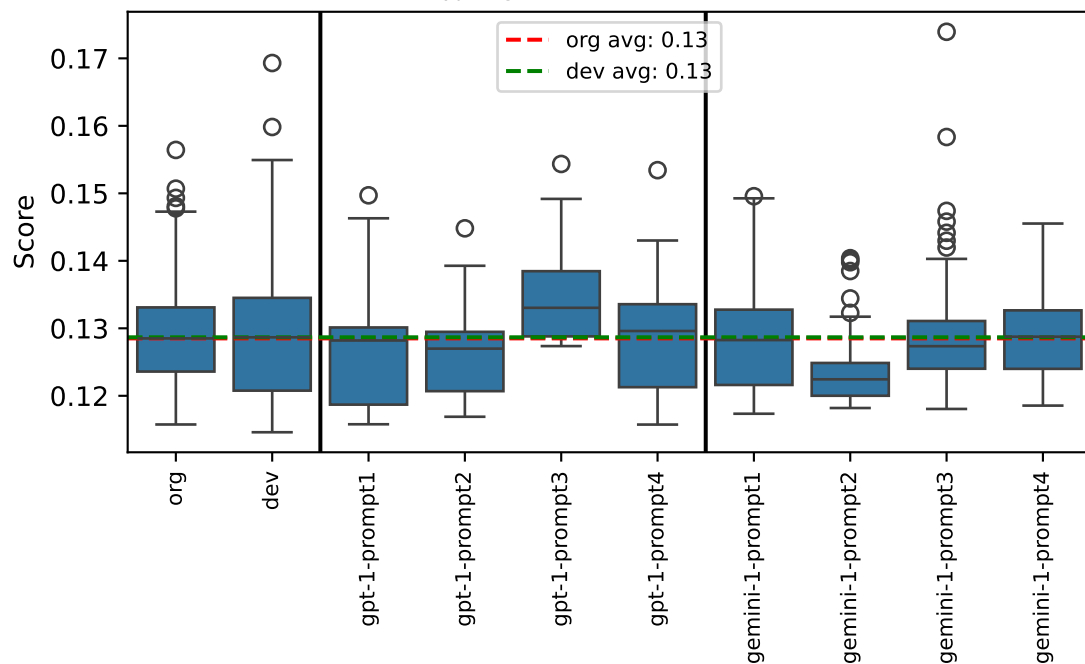


Params: {'typeSignature': 'real', 'withNulls': 'false'}

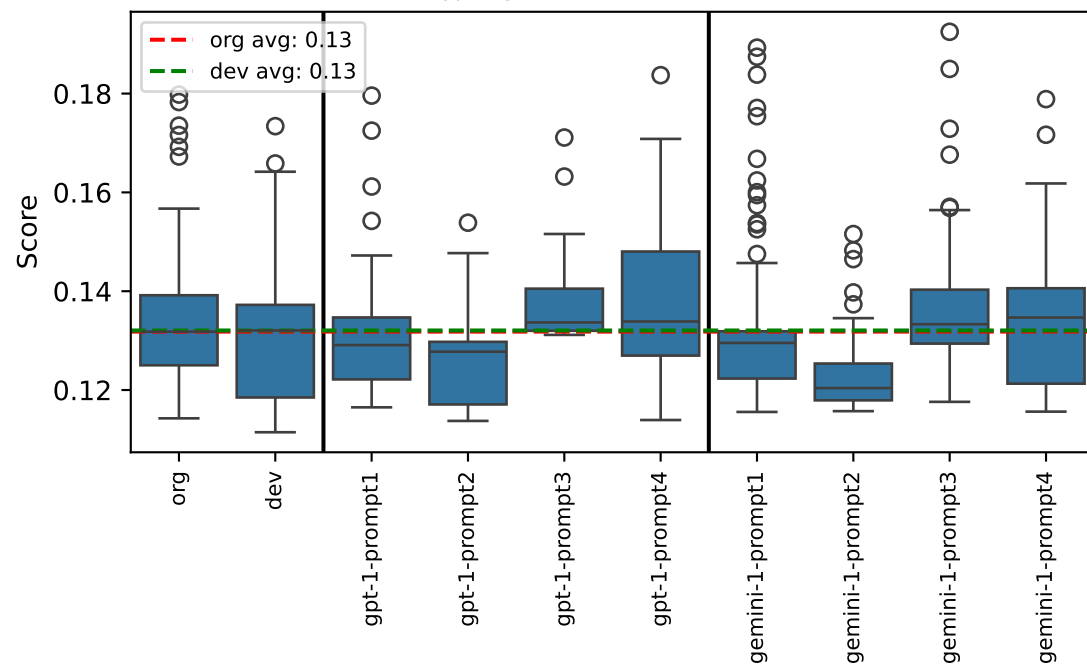


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

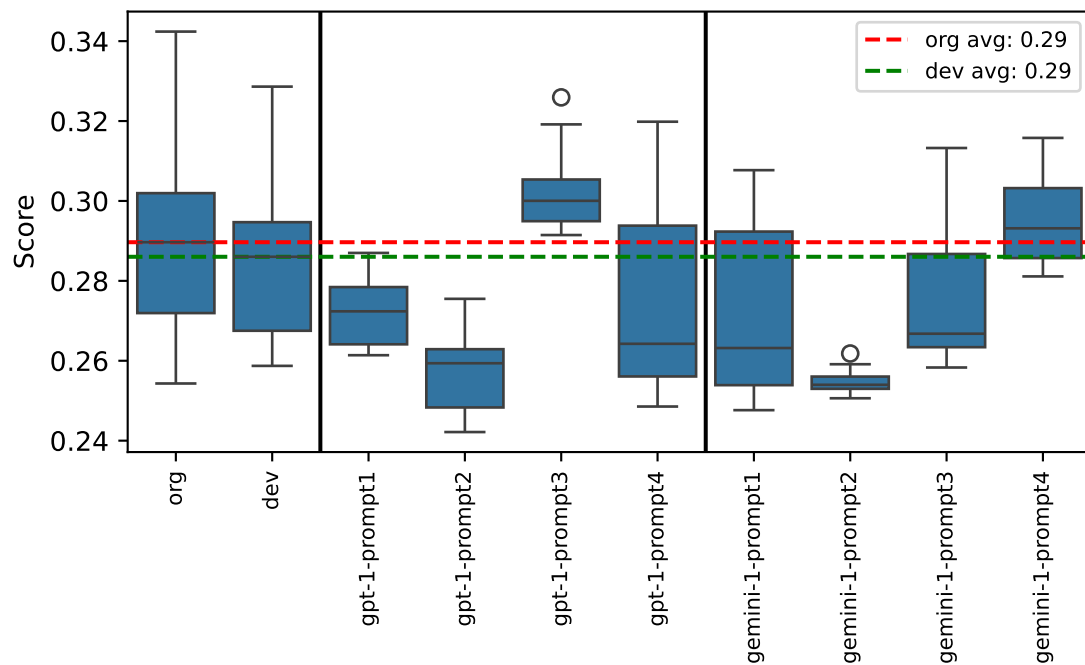
Params: {'typeSignature': 'double', 'withNulls': 'true'}



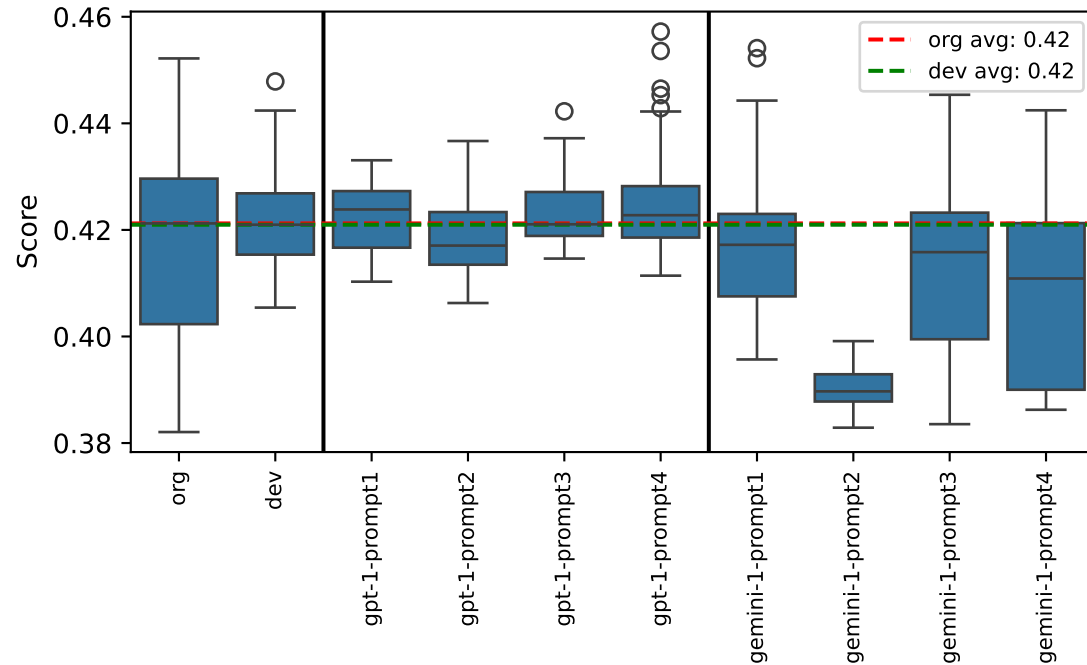
Params: {'typeSignature': 'double', 'withNulls': 'false'}



Params: {'typeSignature': 'decimal(10, 5)', 'withNulls': 'true'}

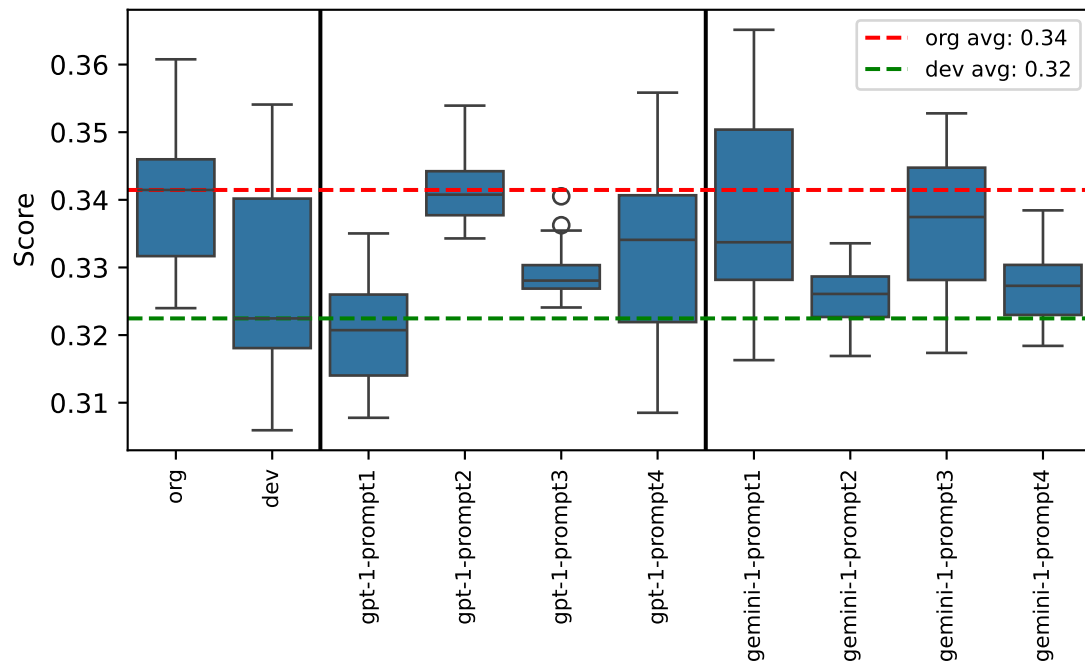


Params: {'typeSignature': 'decimal(10, 5)', 'withNulls': 'false'}

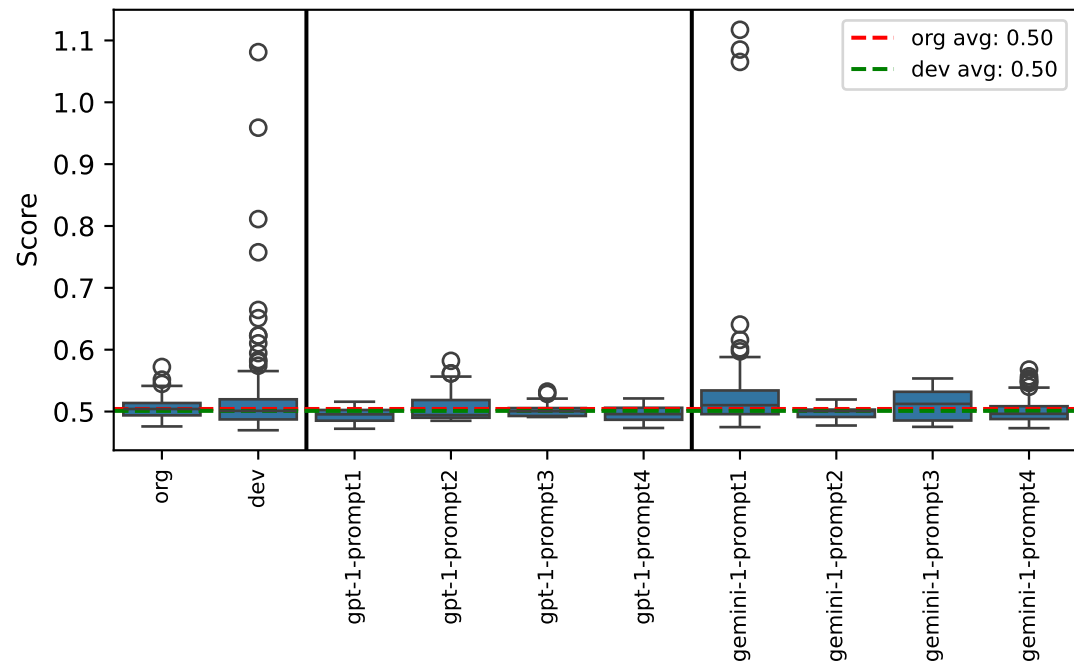


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

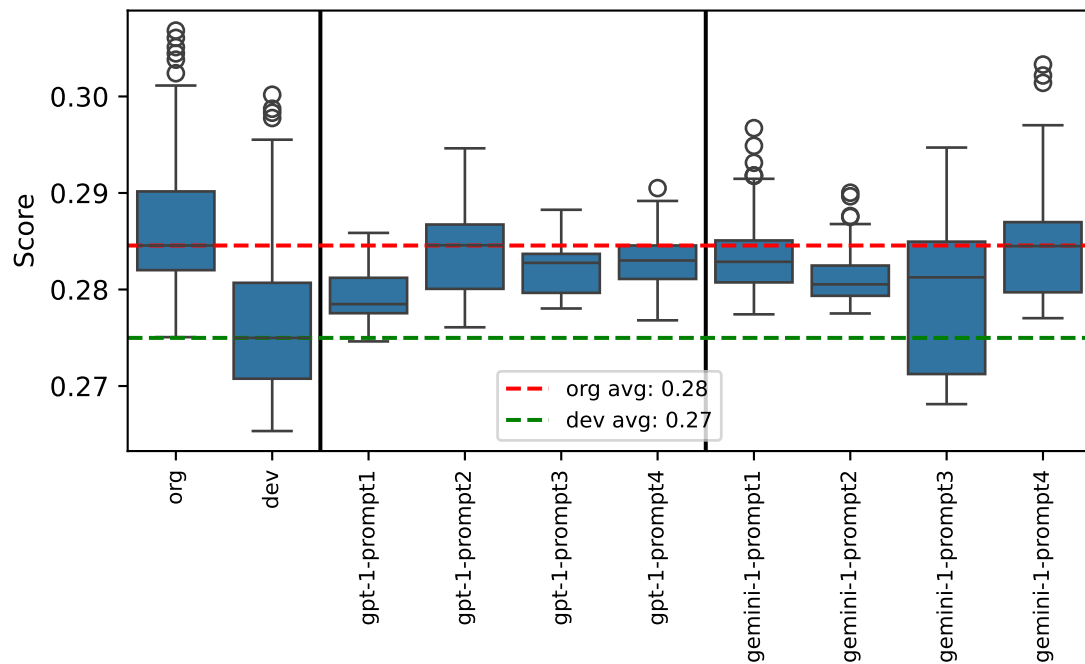
Params: {'typeSignature': 'decimal(30, 10)', 'withNulls': 'true'}



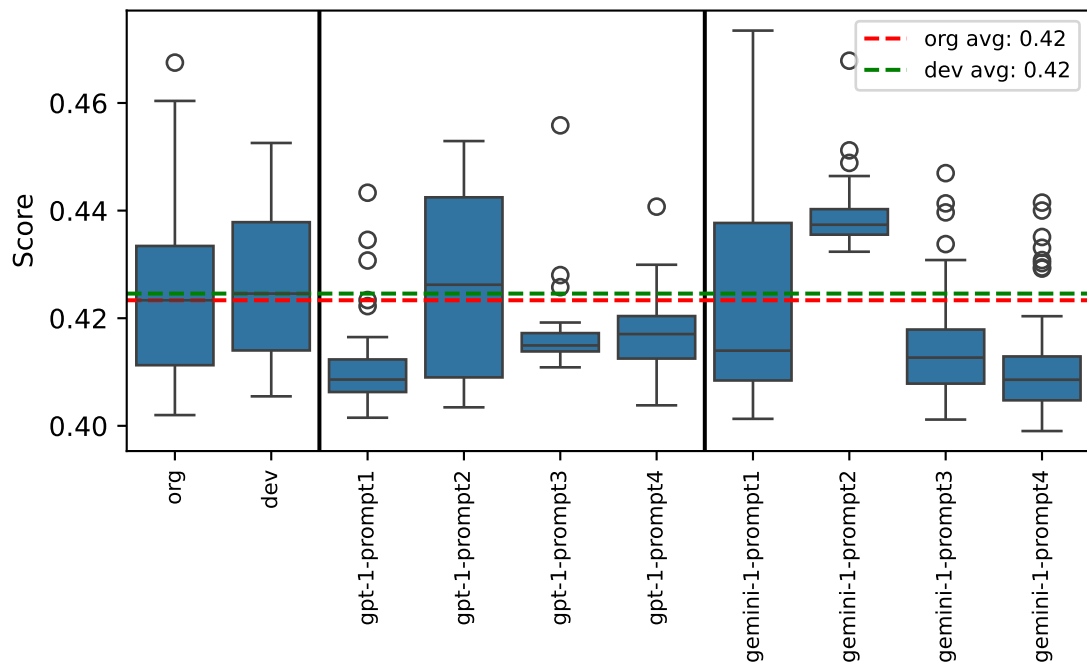
Params: {'typeSignature': 'decimal(30, 10)', 'withNulls': 'false'}



Params: {'typeSignature': 'varchar_direct', 'withNulls': 'true'}

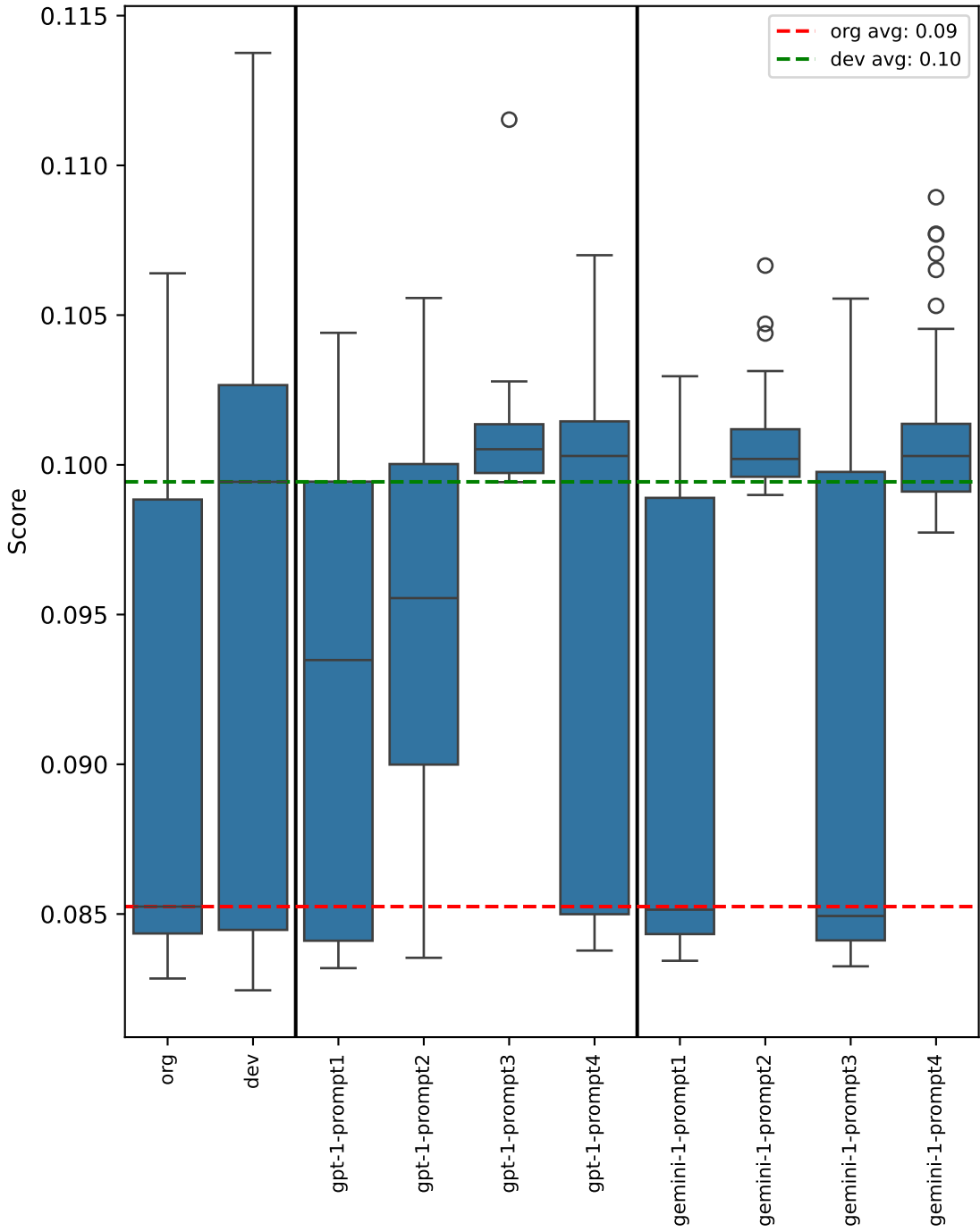


Params: {'typeSignature': 'varchar_direct', 'withNulls': 'false'}

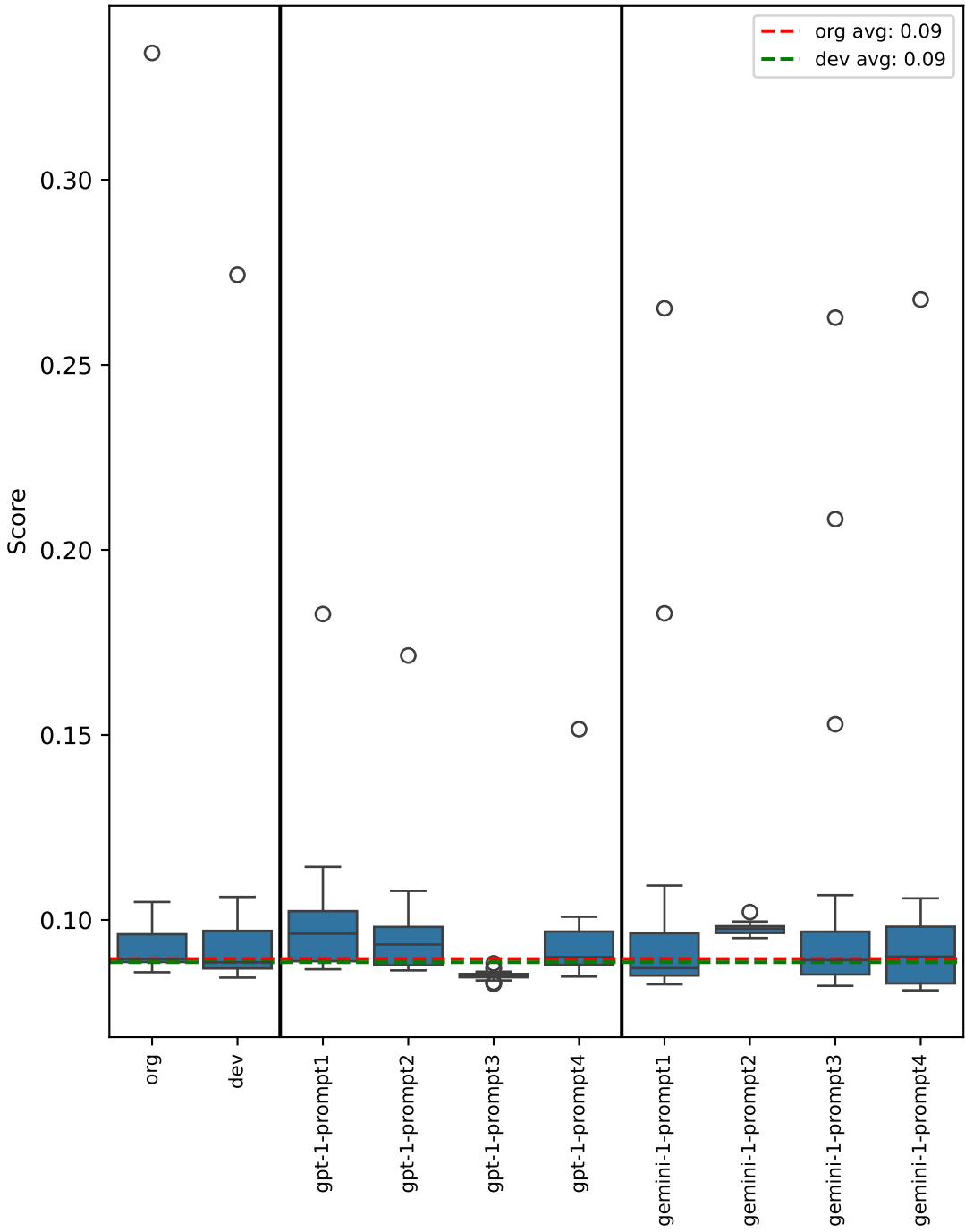


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.read

Params: {'typeSignature': 'varchar_dictionary', 'withNulls': 'true'}

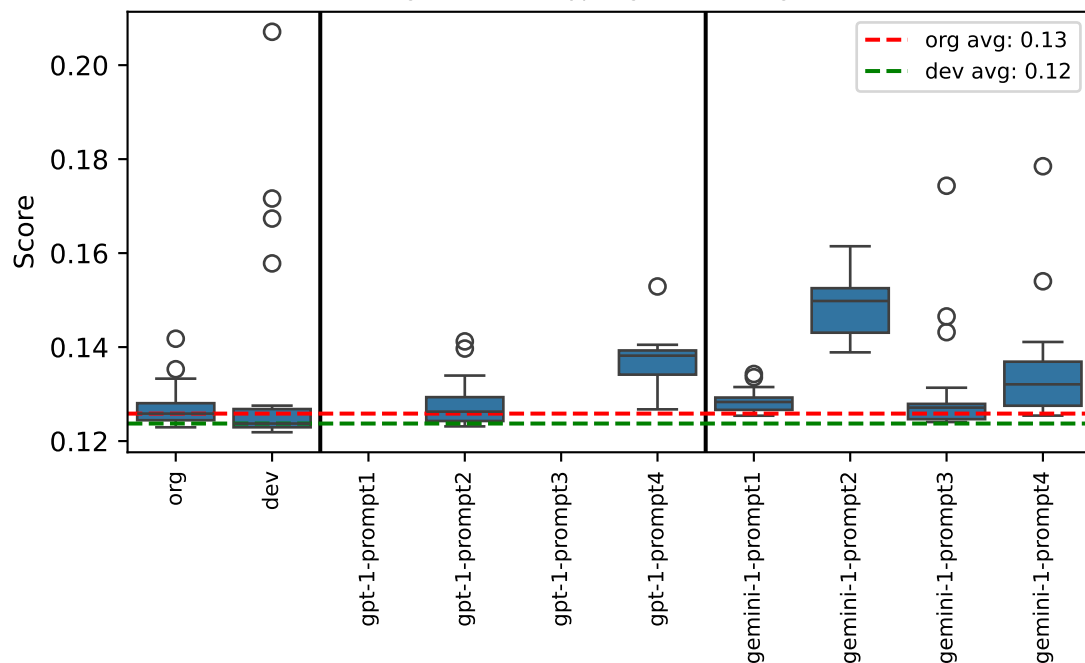


Params: {'typeSignature': 'varchar_dictionary', 'withNulls': 'false'}

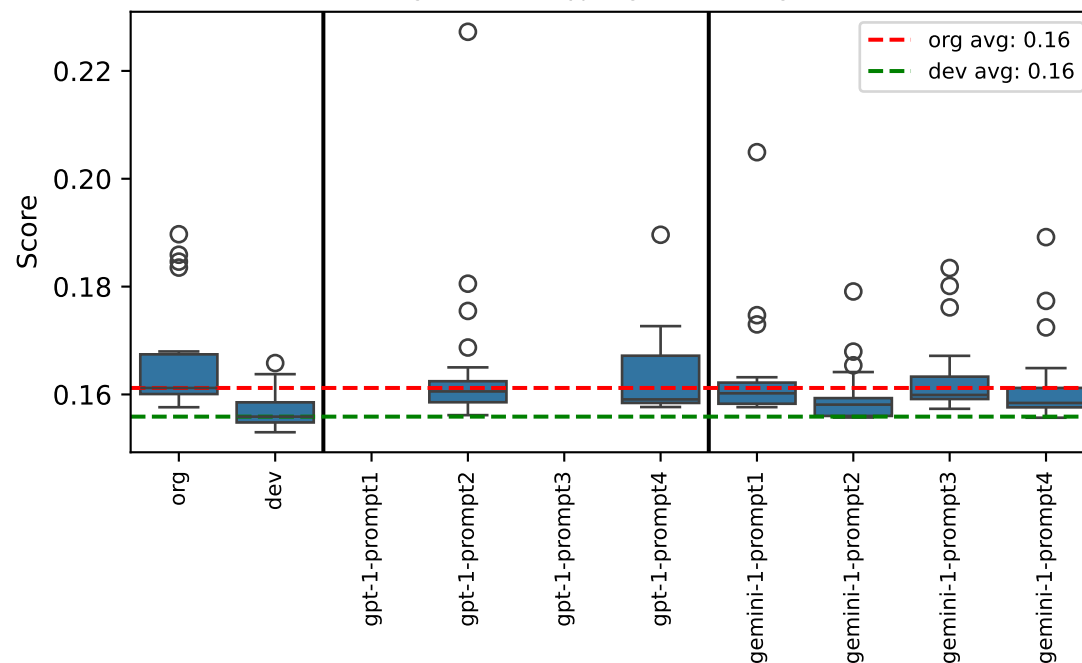


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

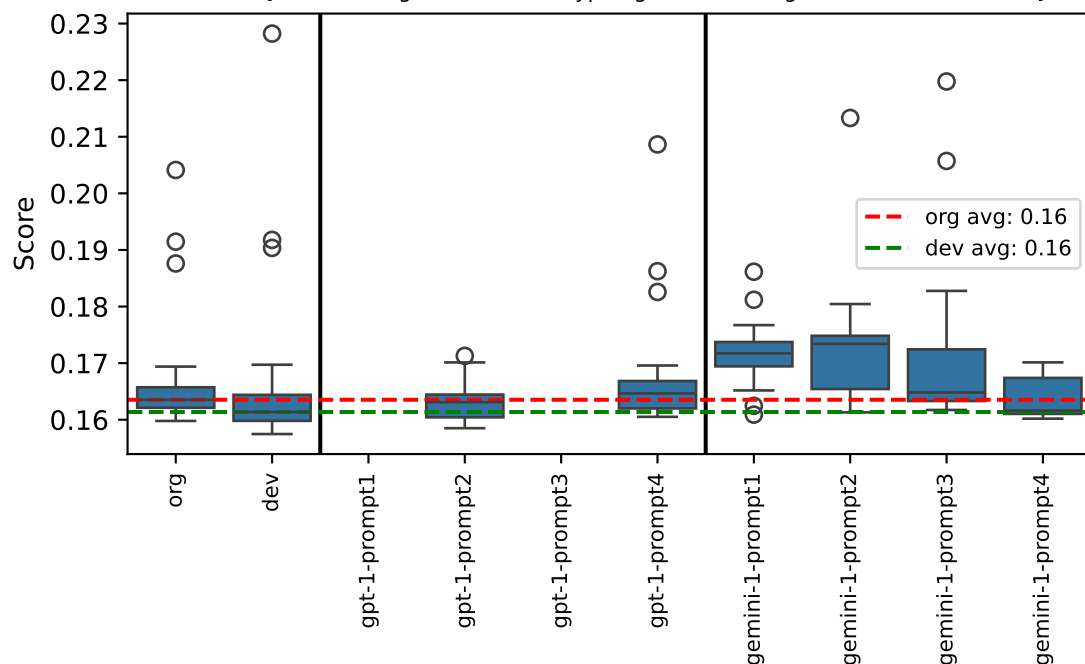
Params: {'filterRateSignature': '-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



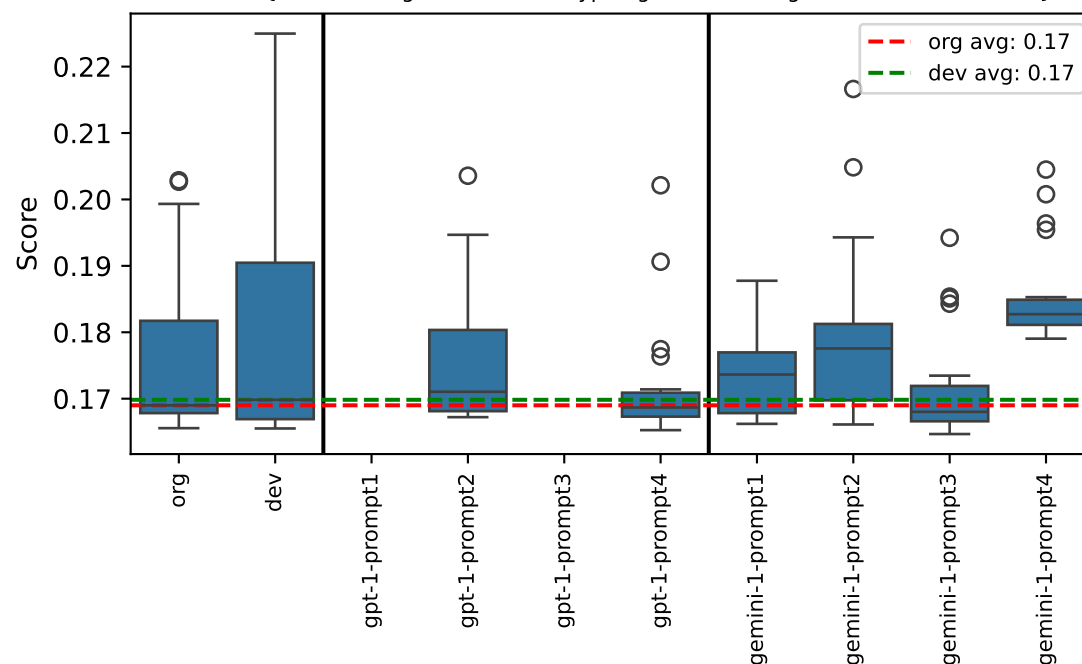
Params: {'filterRateSignature': '0', 'typeSignature': 'integer', 'withNulls': 'NONE'}



Params: {'filterRateSignature': '0.1', 'typeSignature': 'integer', 'withNulls': 'NONE'}

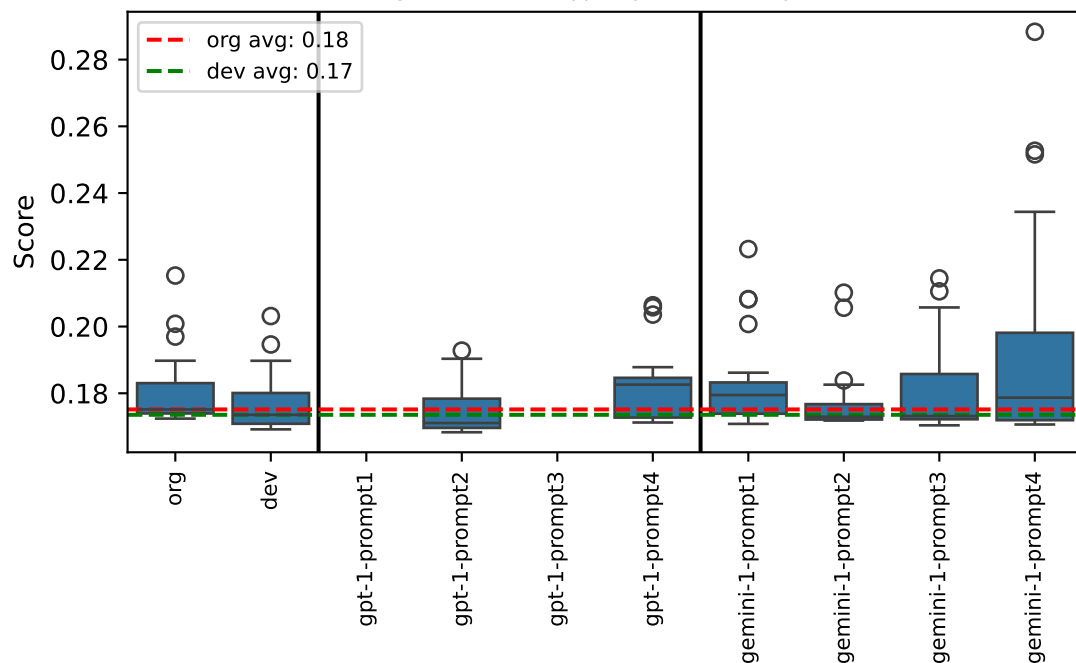


Params: {'filterRateSignature': '0.2', 'typeSignature': 'integer', 'withNulls': 'NONE'}

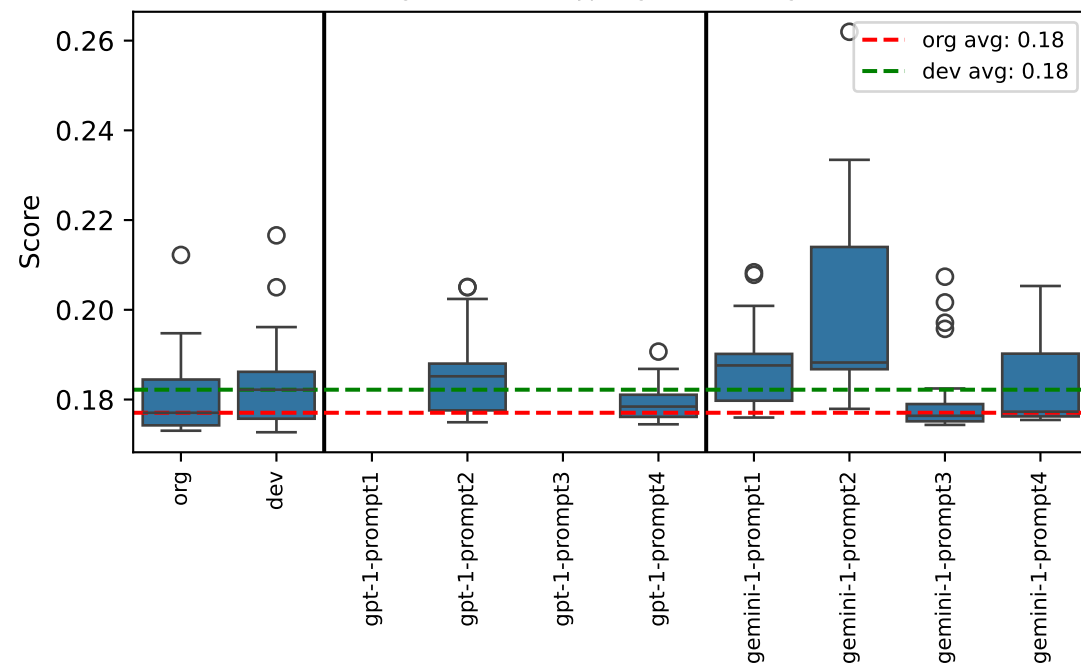


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

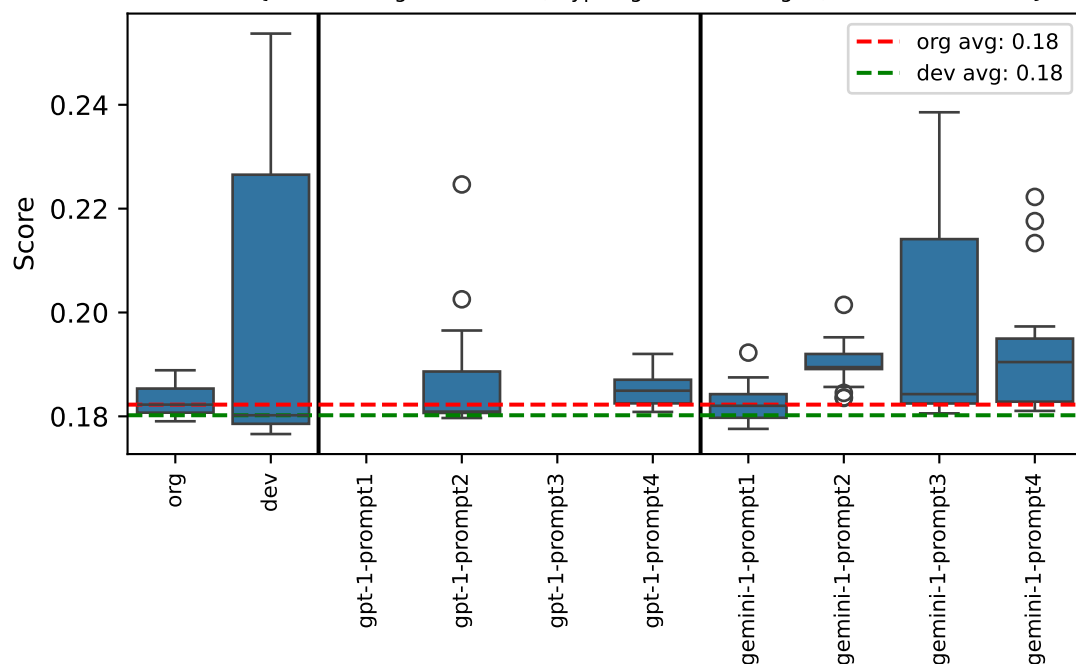
Params: {'filterRateSignature': '0.3', 'typeSignature': 'integer', 'withNulls': 'NONE'}



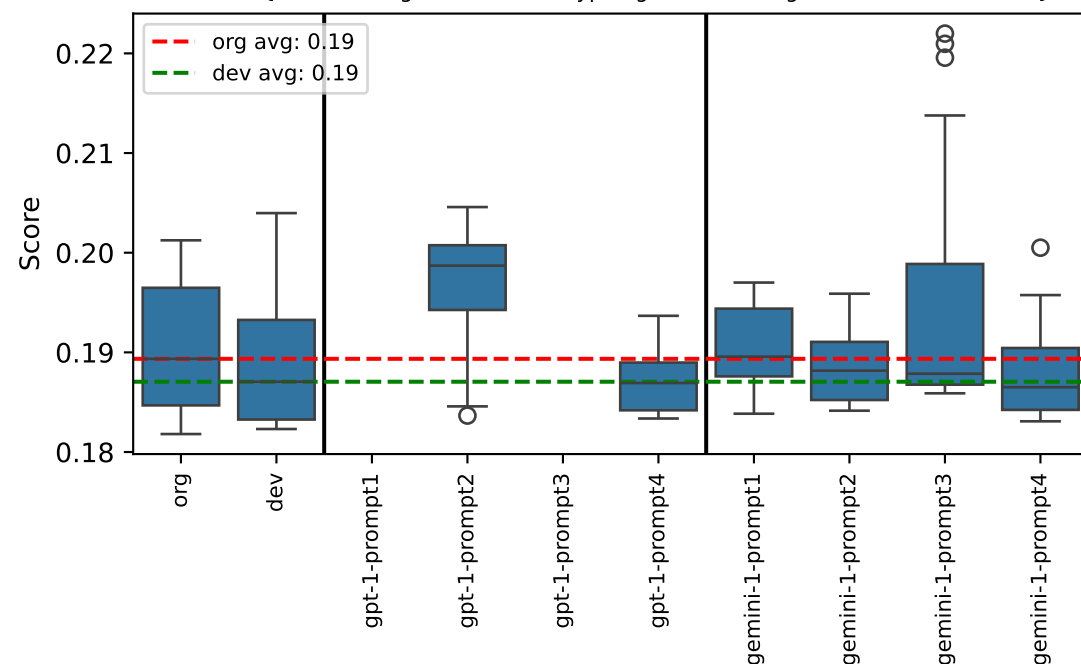
Params: {'filterRateSignature': '0.4', 'typeSignature': 'integer', 'withNulls': 'NONE'}



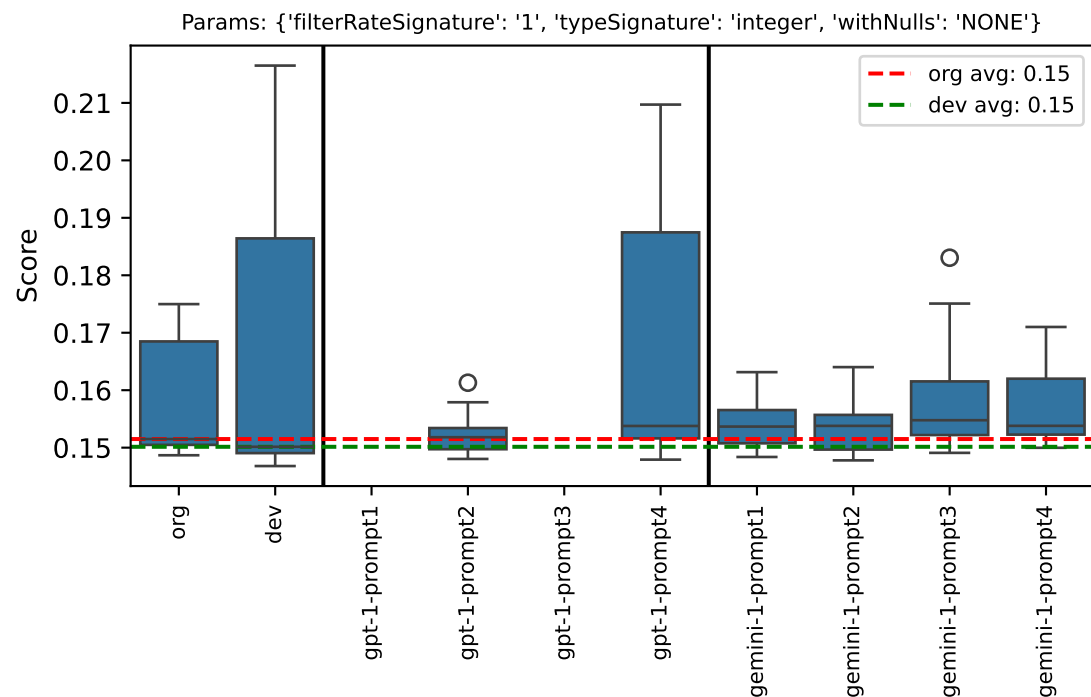
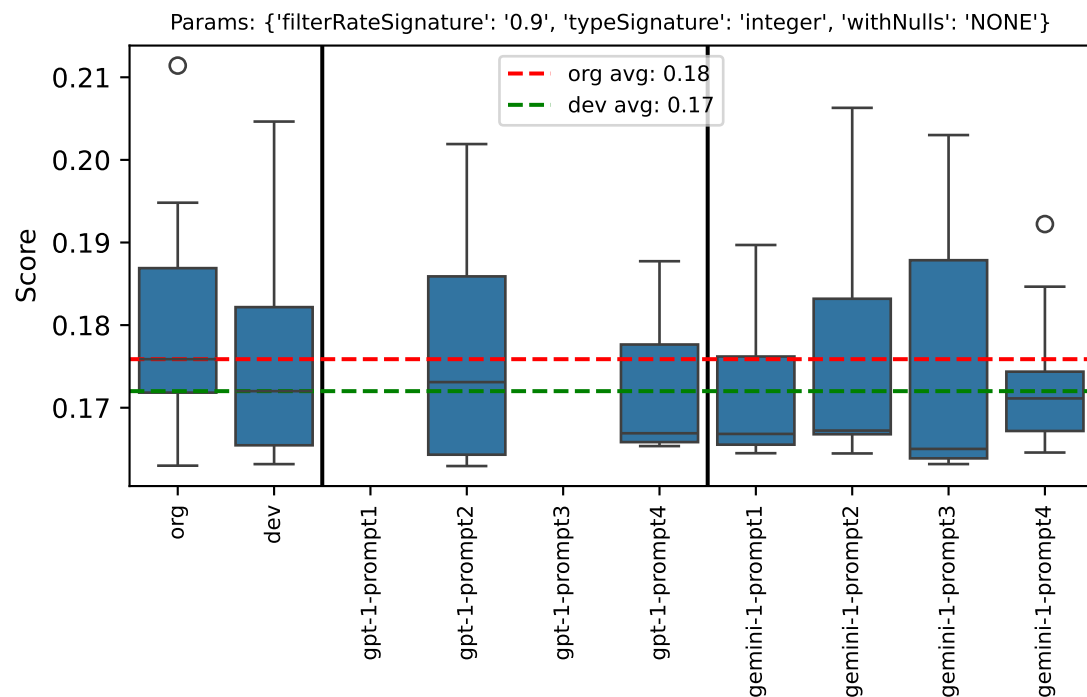
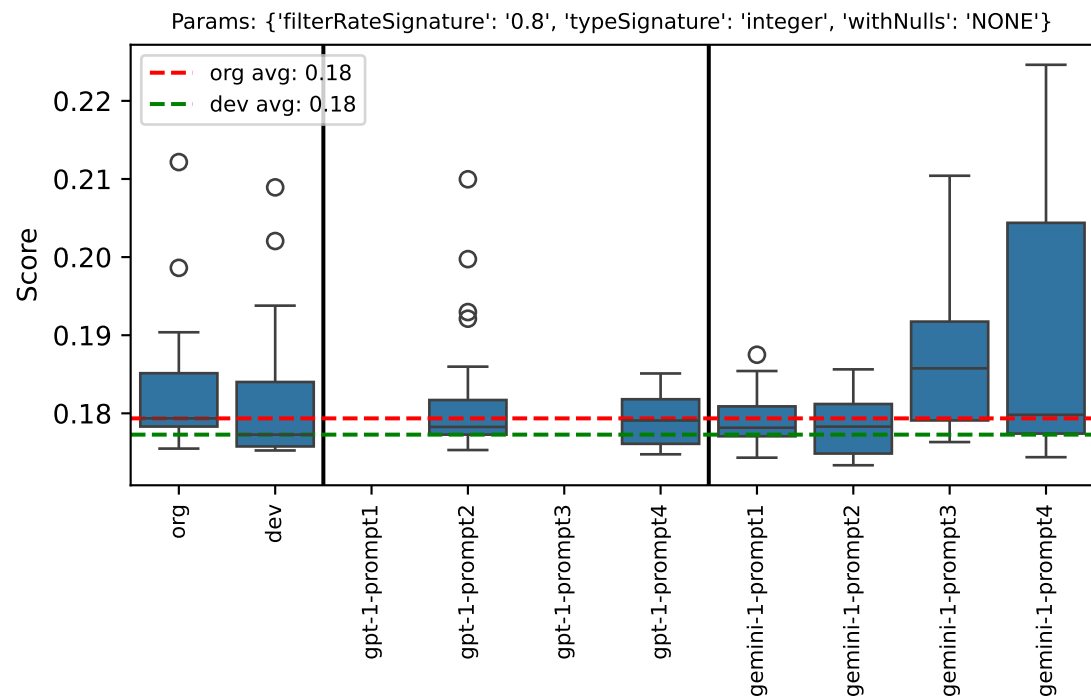
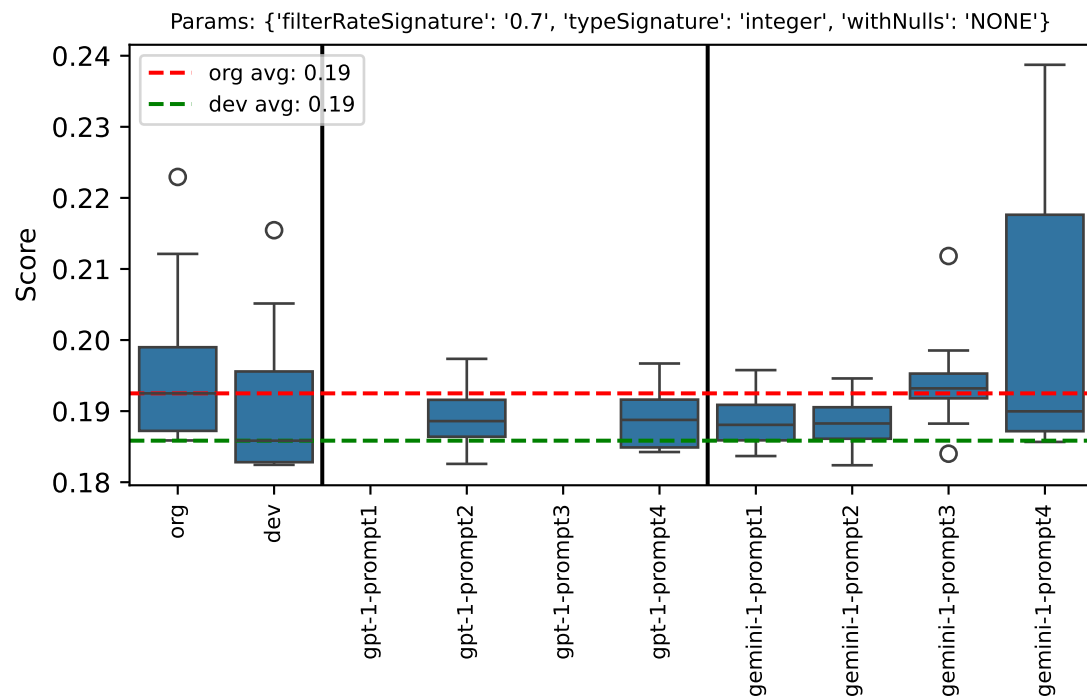
Params: {'filterRateSignature': '0.5', 'typeSignature': 'integer', 'withNulls': 'NONE'}



Params: {'filterRateSignature': '0.6', 'typeSignature': 'integer', 'withNulls': 'NONE'}

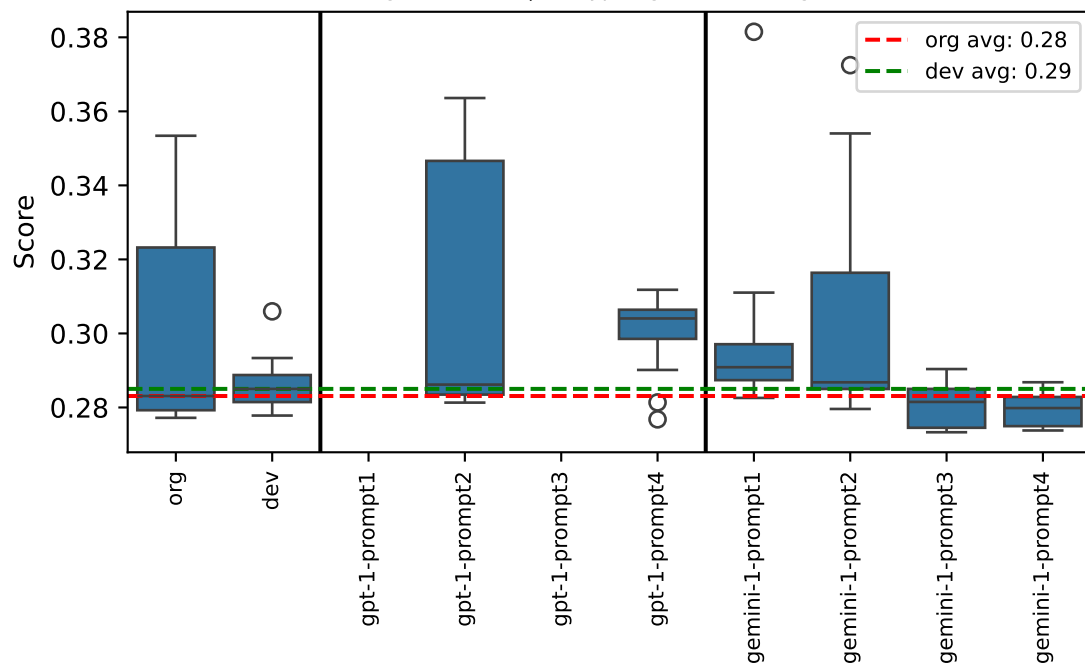


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

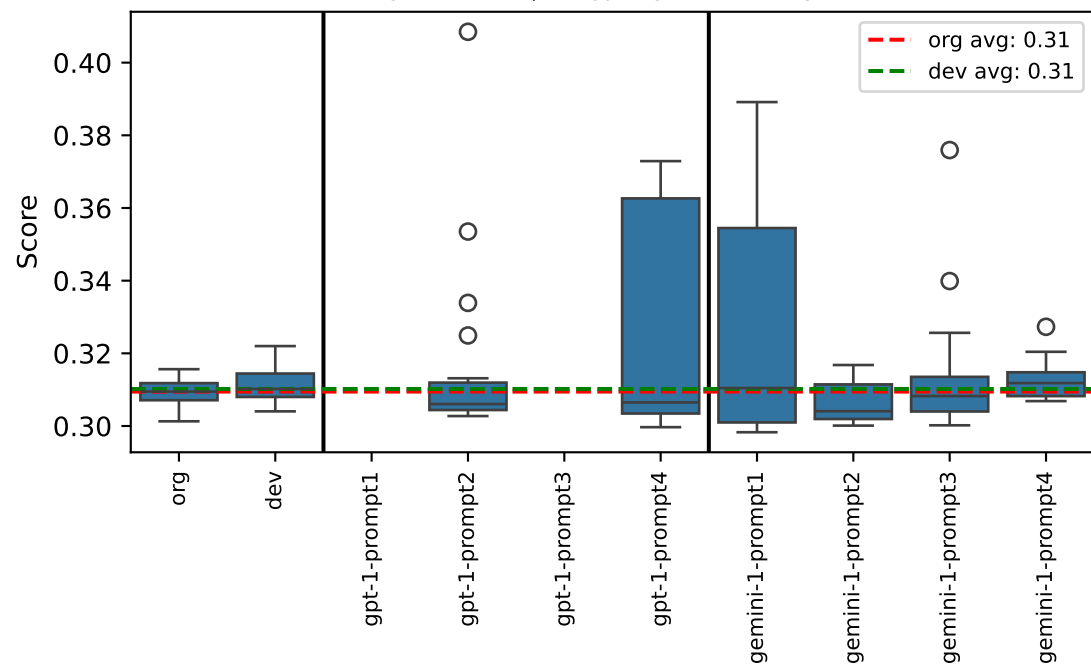


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

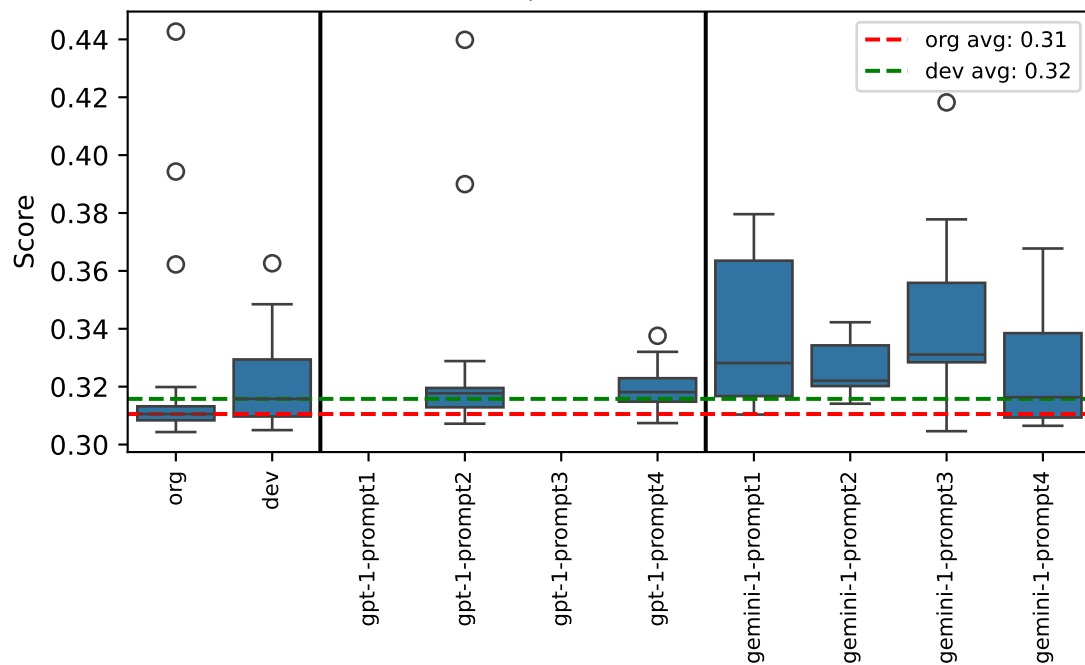
Params: {'filterRateSignature': '0.0|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



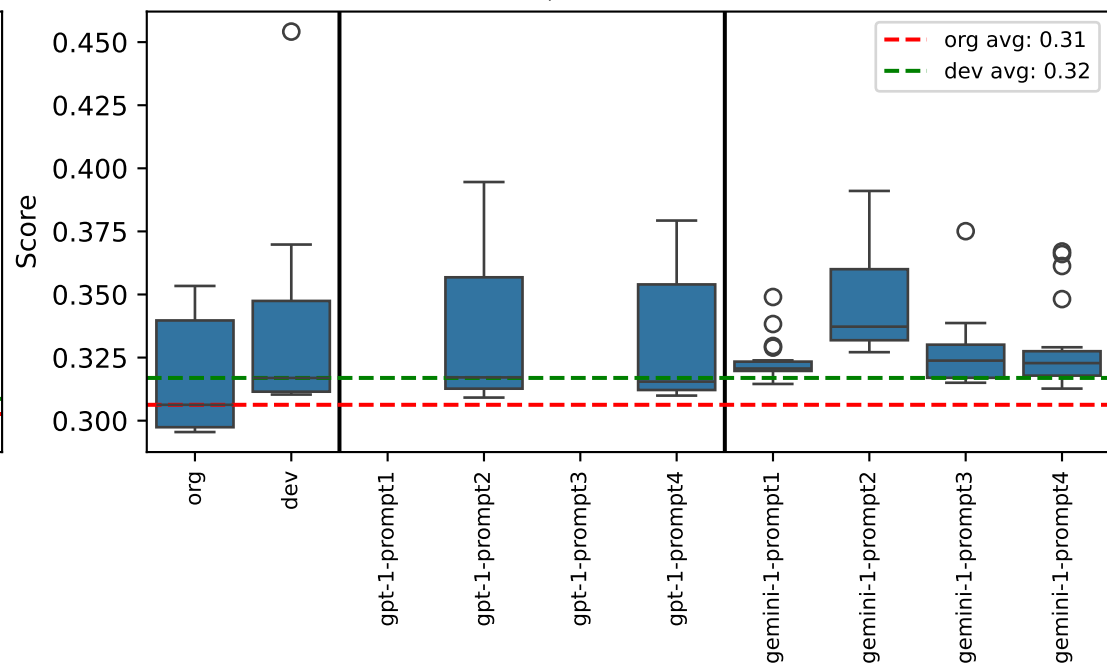
Params: {'filterRateSignature': '0.1|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



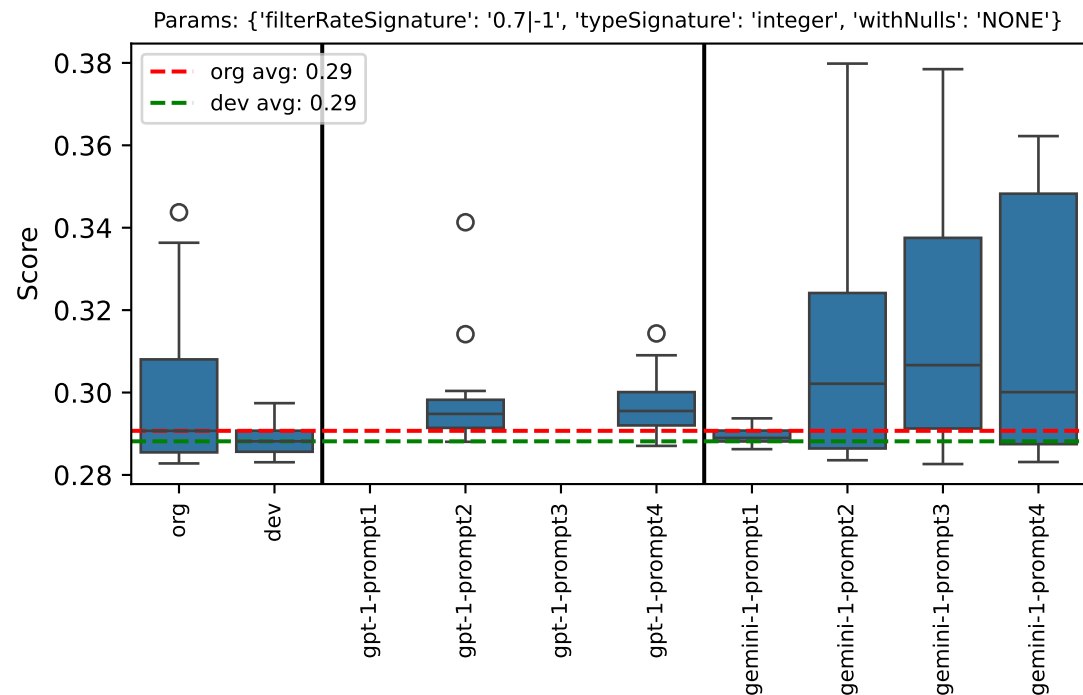
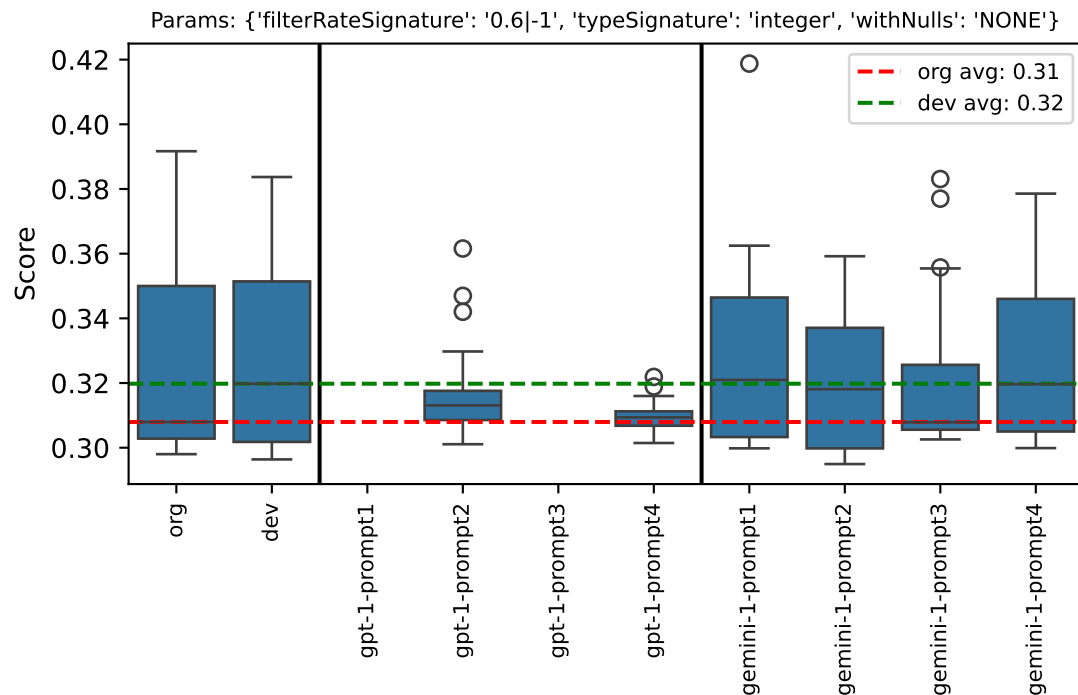
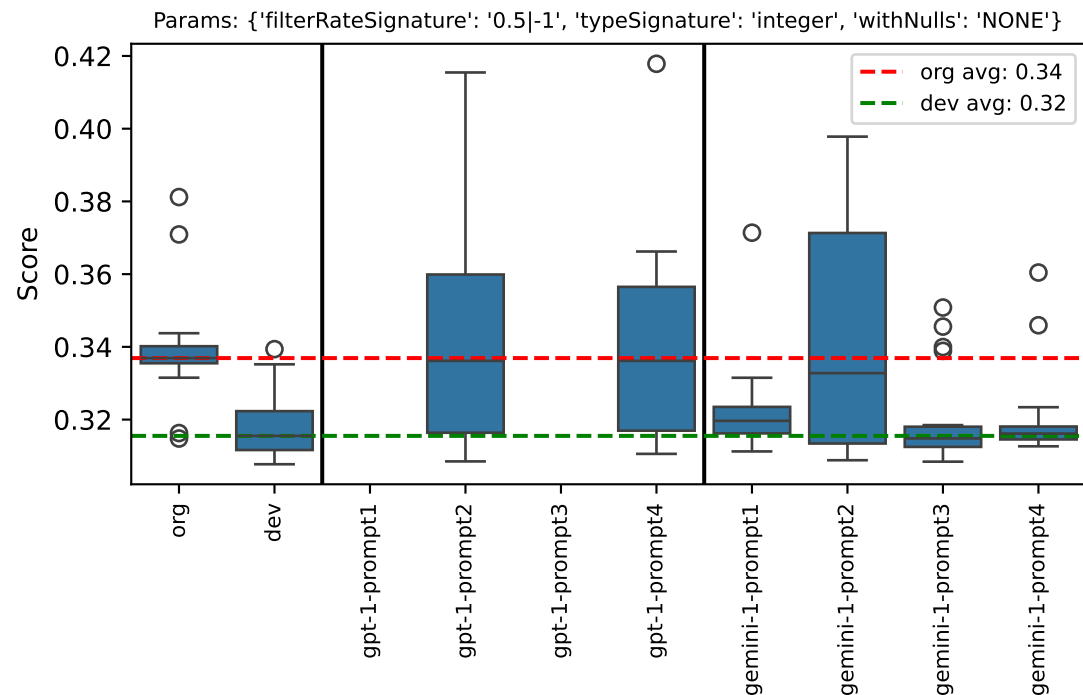
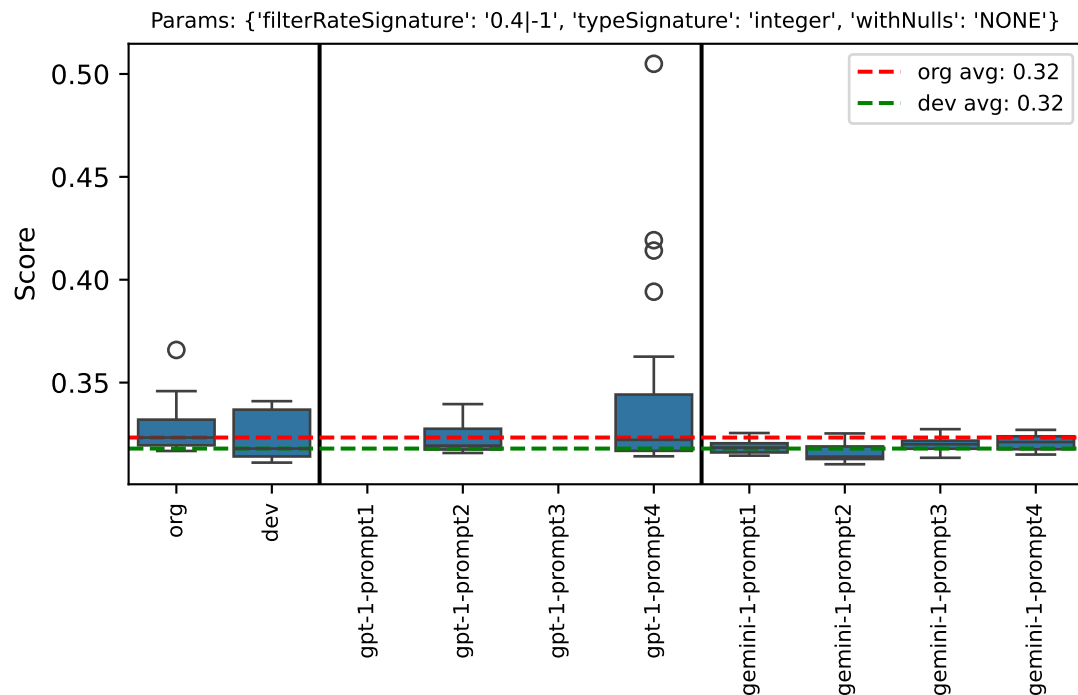
Params: {'filterRateSignature': '0.2|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



Params: {'filterRateSignature': '0.3|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}

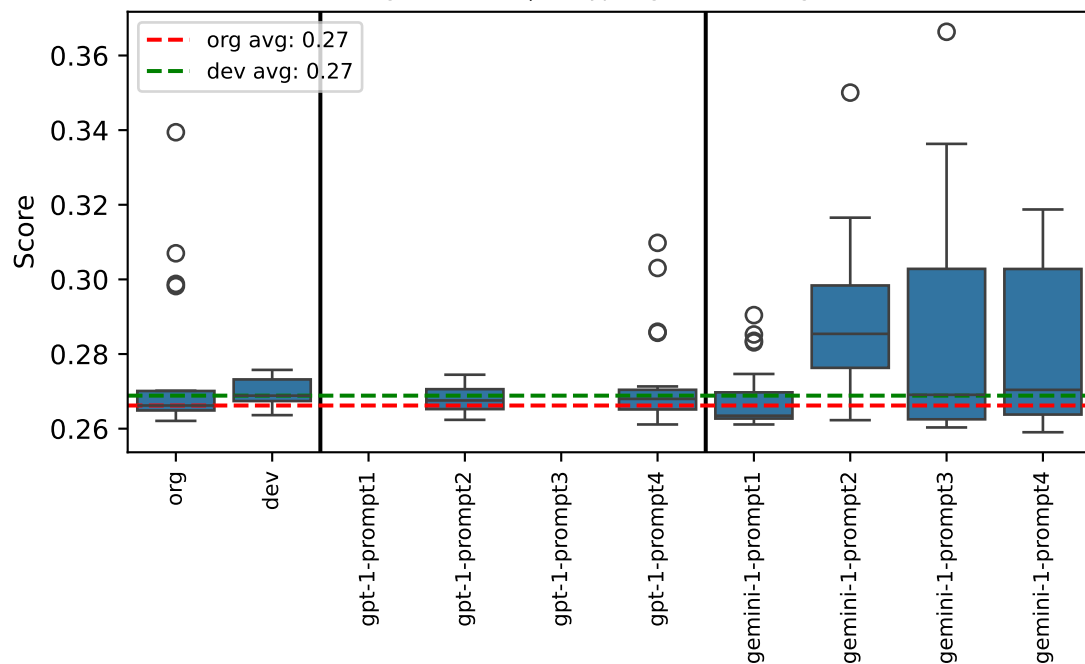


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

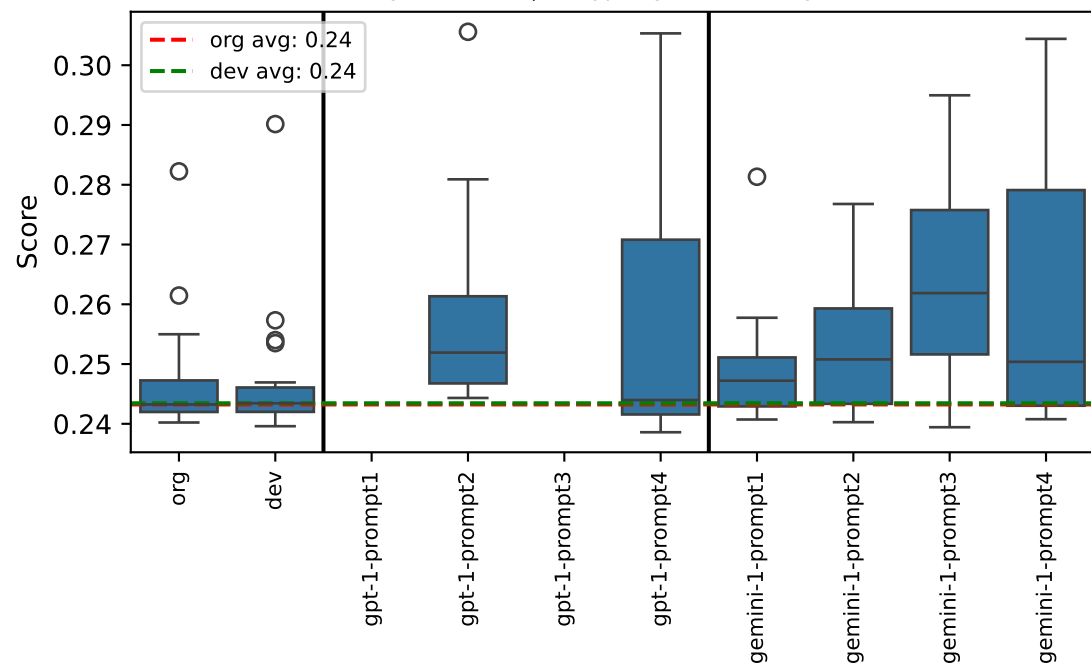


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllBlocks

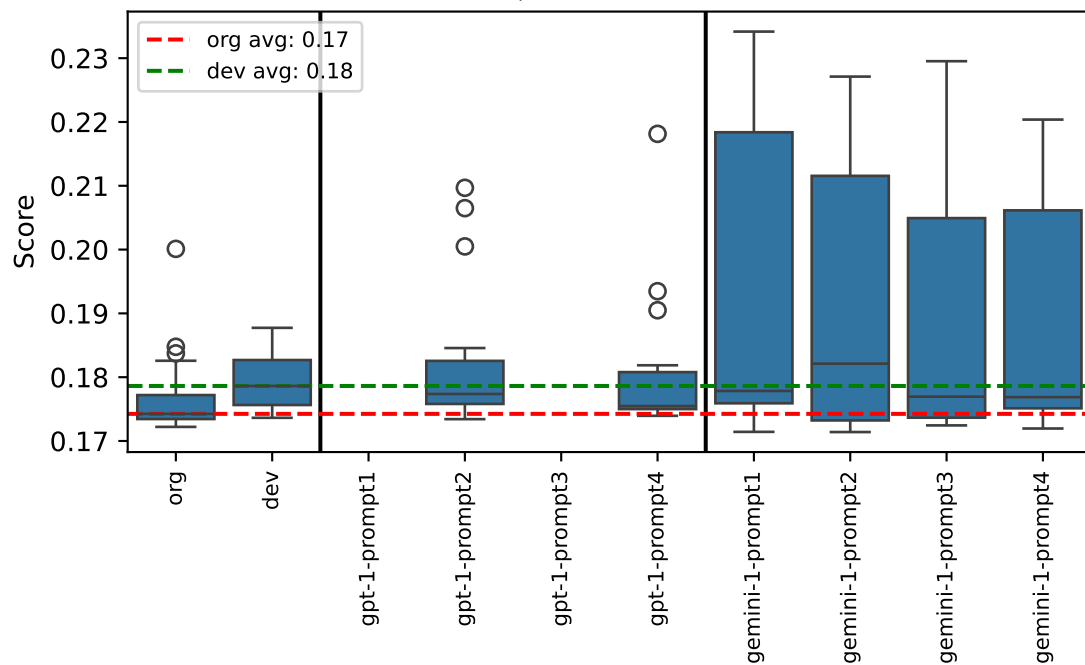
Params: {'filterRateSignature': '0.8|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



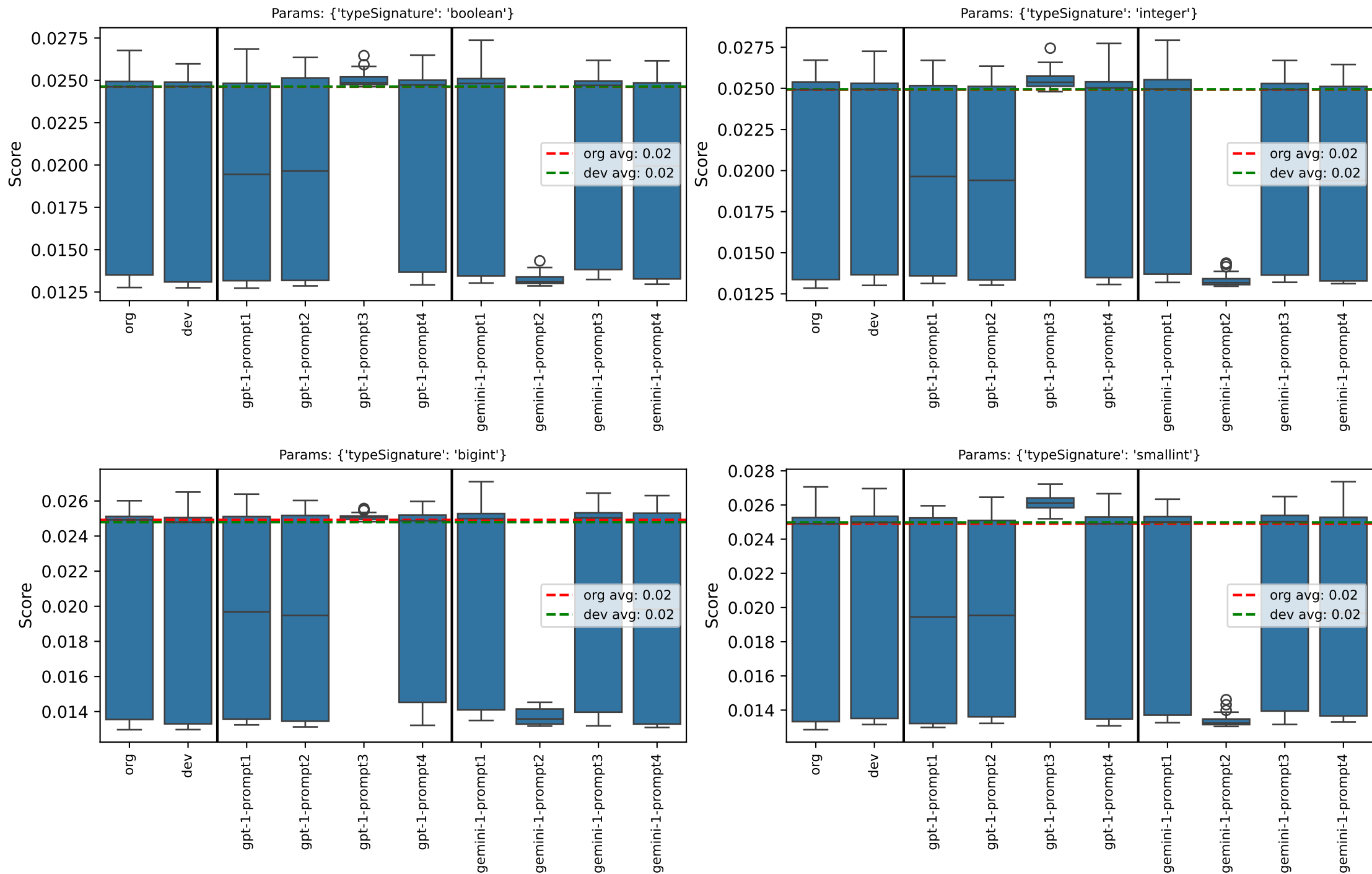
Params: {'filterRateSignature': '0.9|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}



Params: {'filterRateSignature': '1|-1', 'typeSignature': 'integer', 'withNulls': 'NONE'}

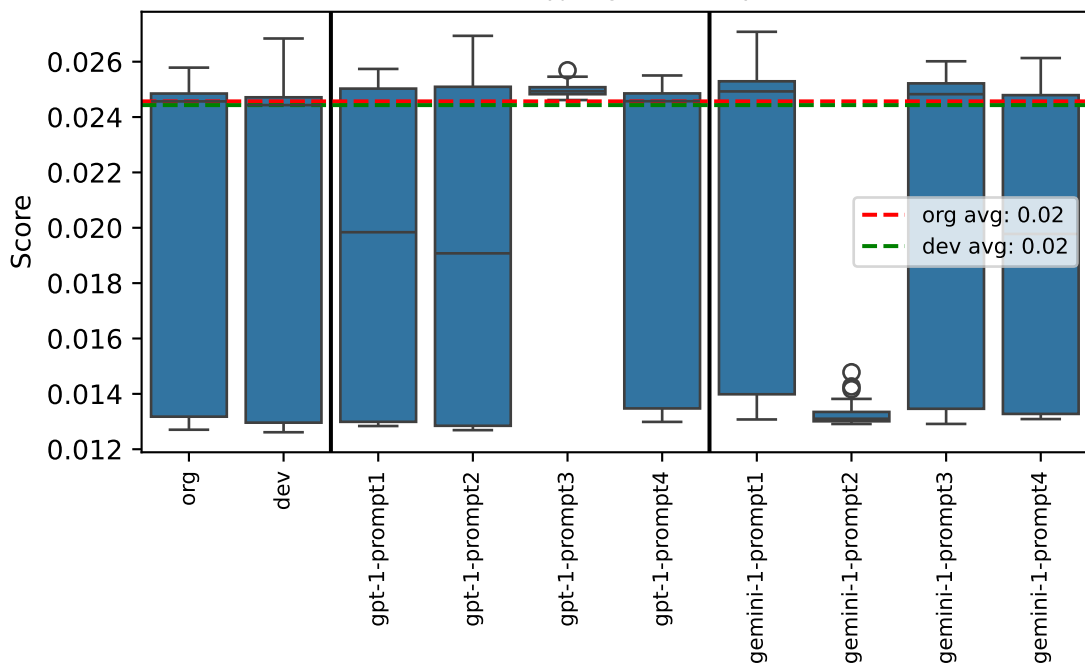


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllNull

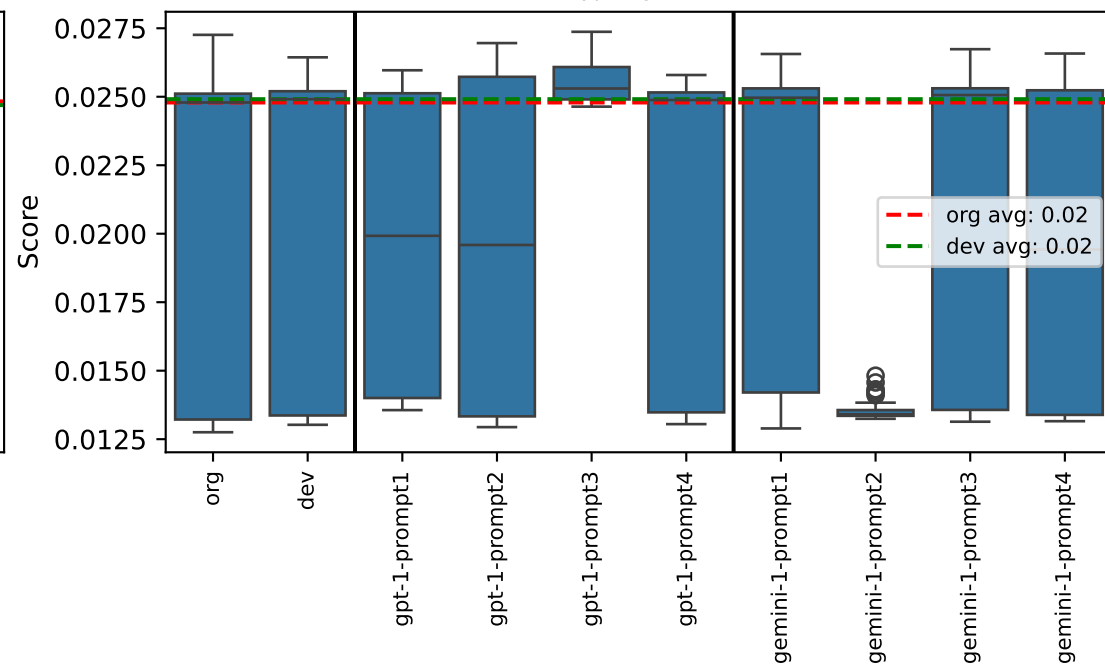


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllNull

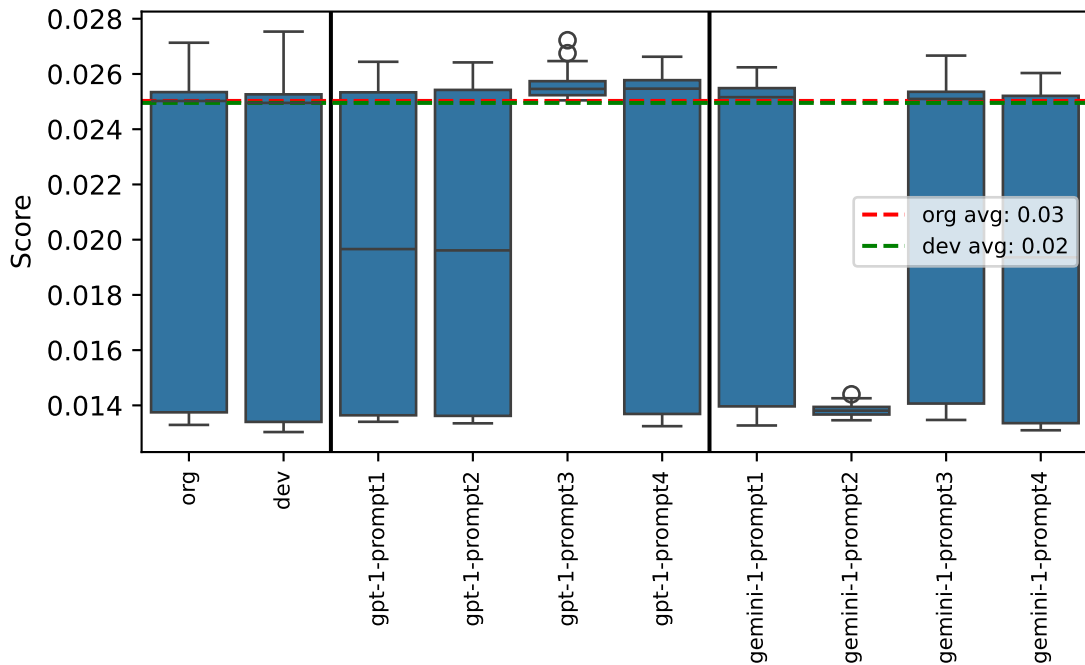
Params: {'typeSignature': 'tinyint'}



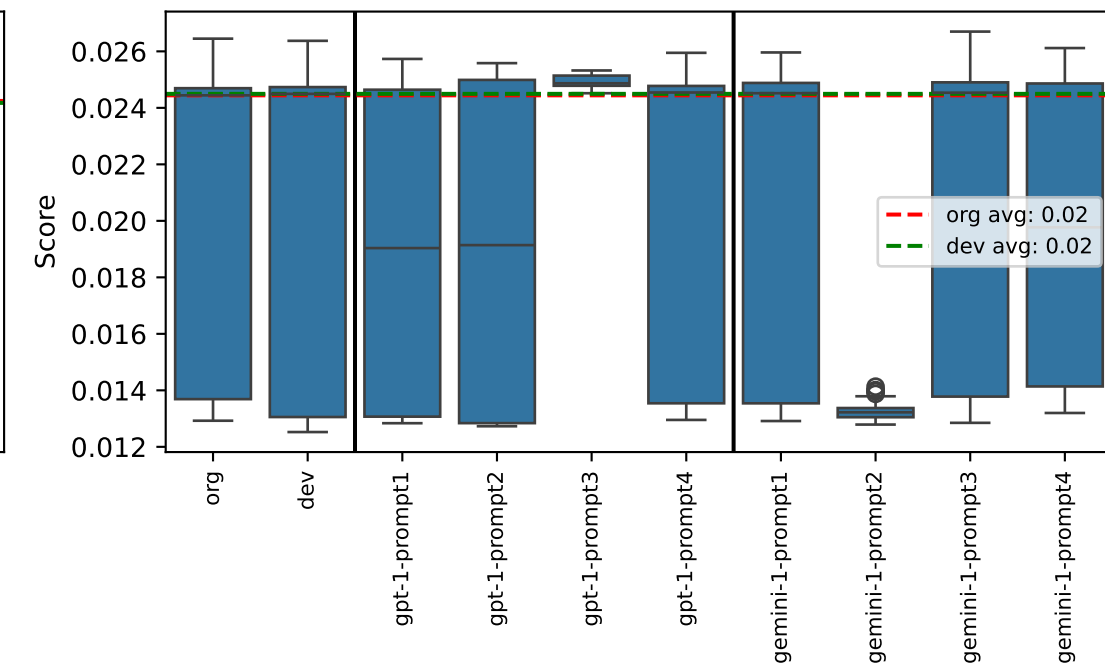
Params: {'typeSignature': 'date'}



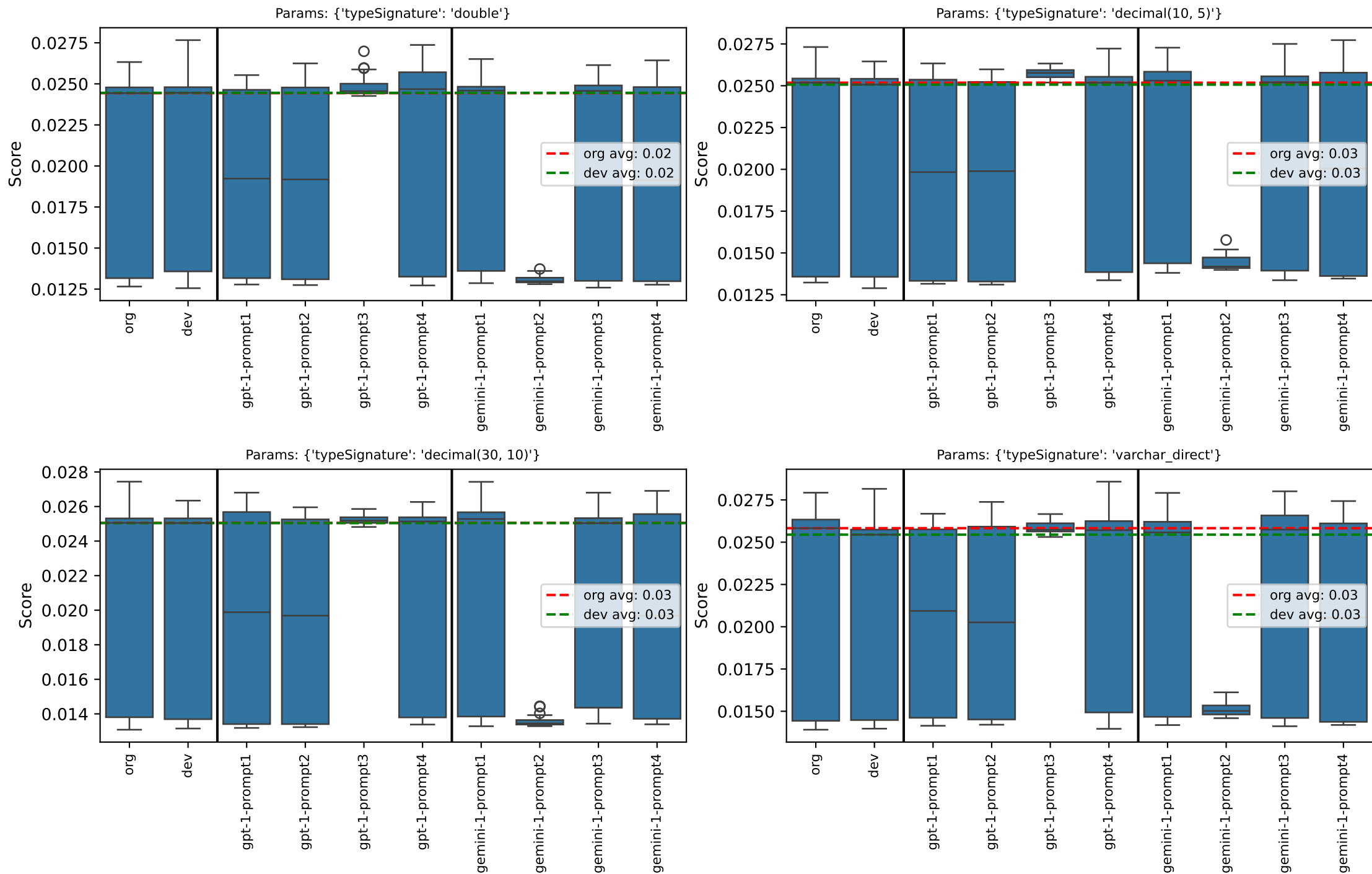
Params: {'typeSignature': 'timestamp'}



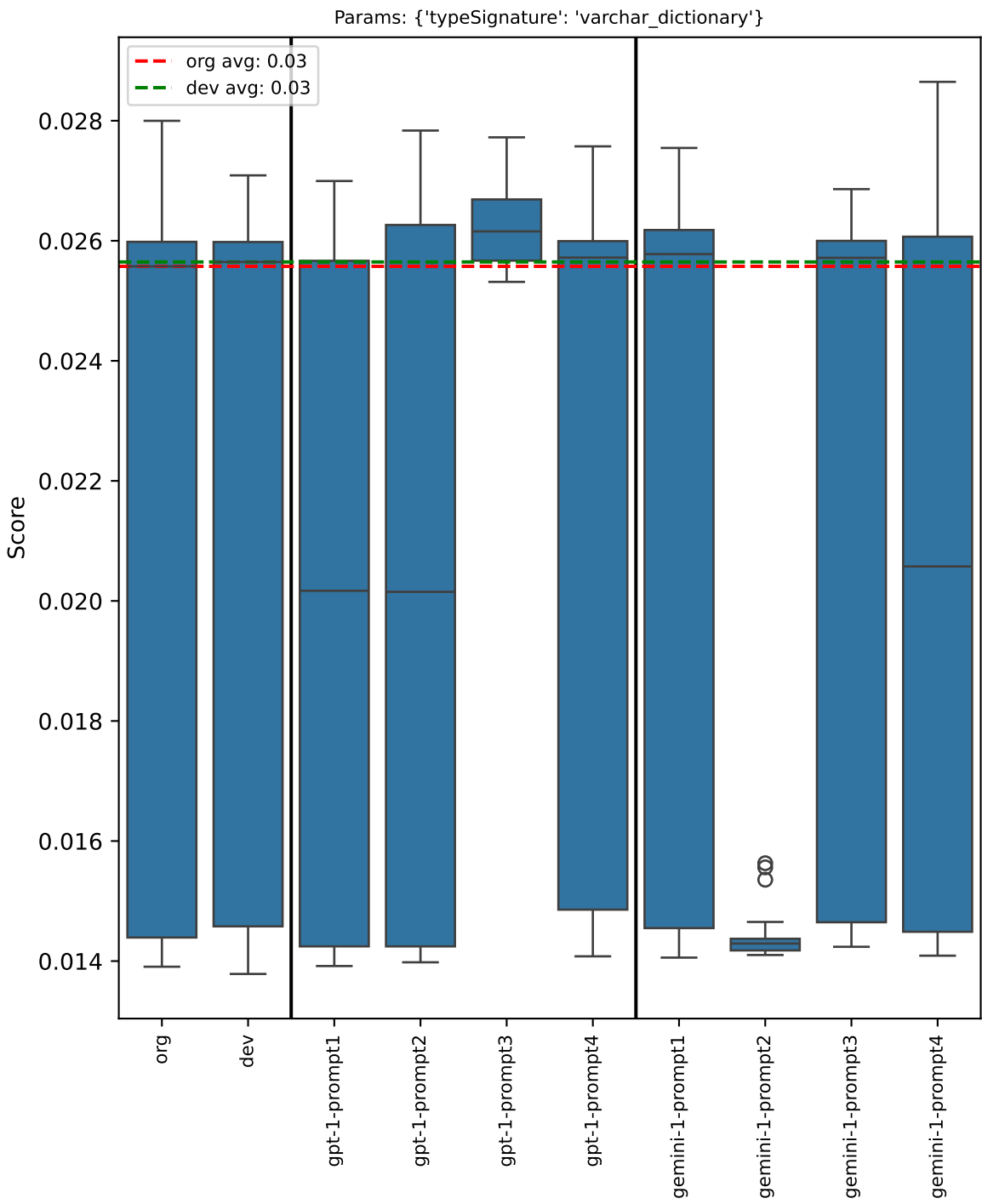
Params: {'typeSignature': 'real'}



Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllNull

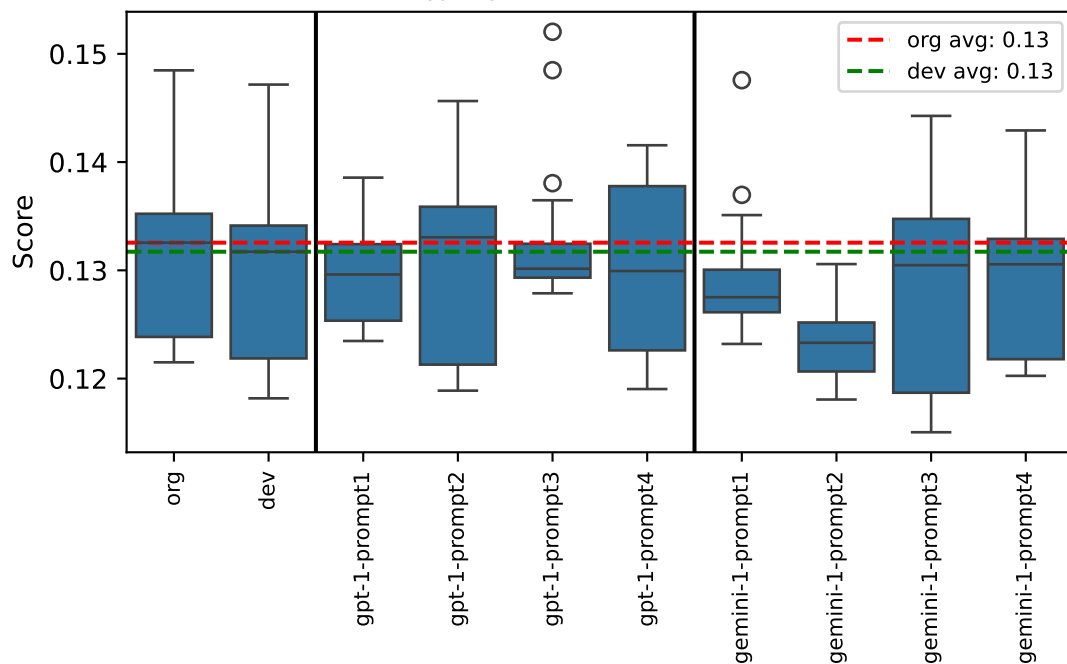


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readAllNull

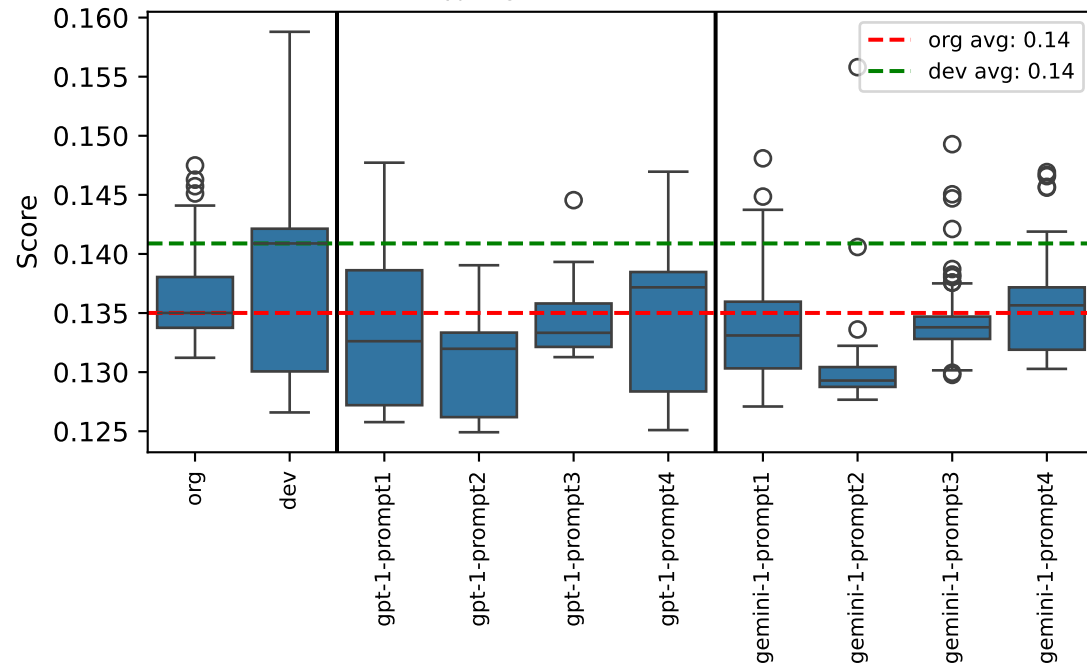


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

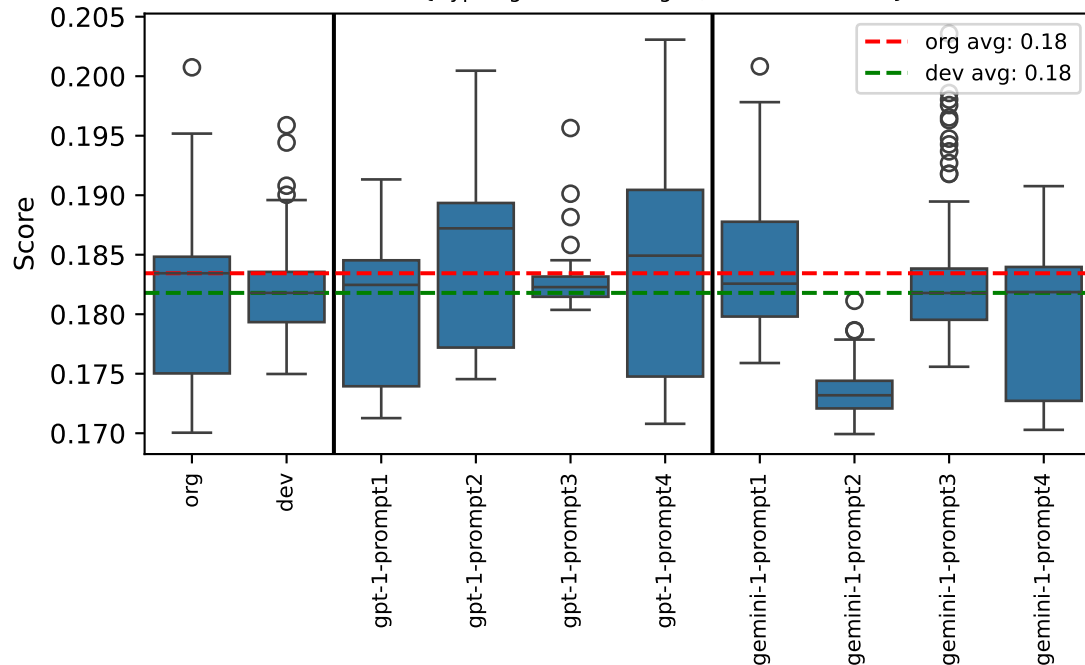
Params: {'typeSignature': 'boolean', 'withNulls': 'true'}



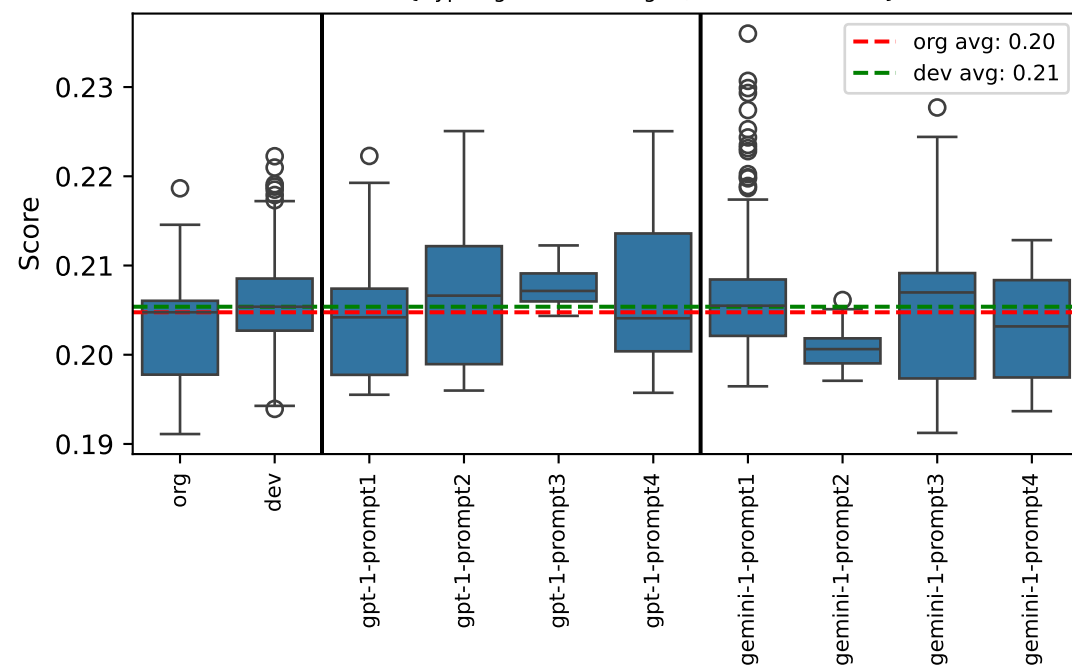
Params: {'typeSignature': 'boolean', 'withNulls': 'false'}



Params: {'typeSignature': 'integer', 'withNulls': 'true'}

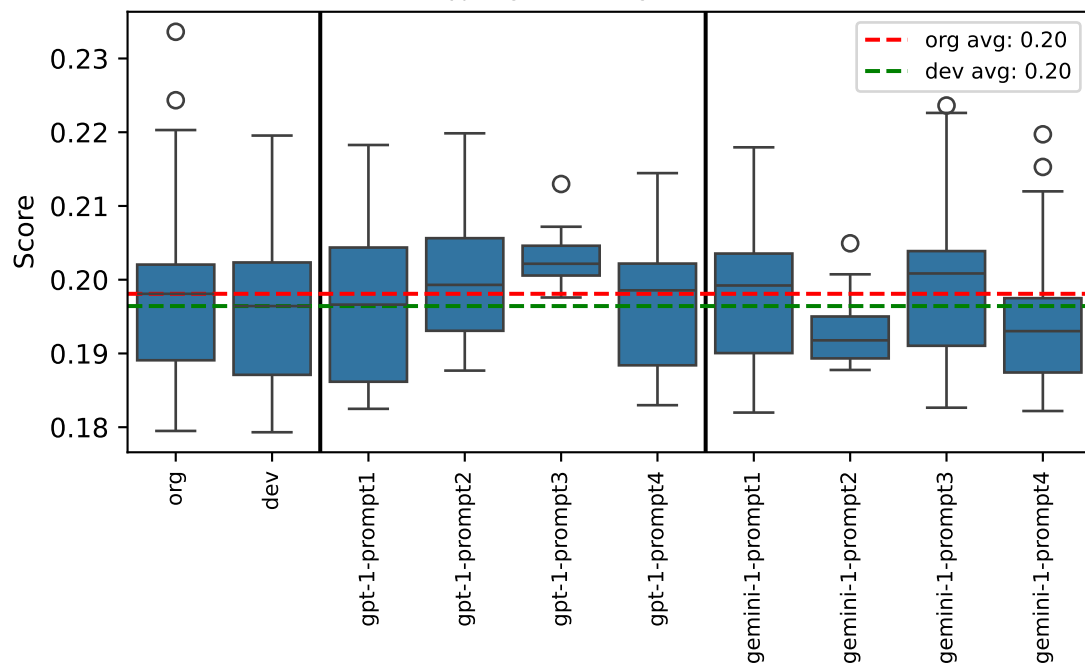


Params: {'typeSignature': 'integer', 'withNulls': 'false'}

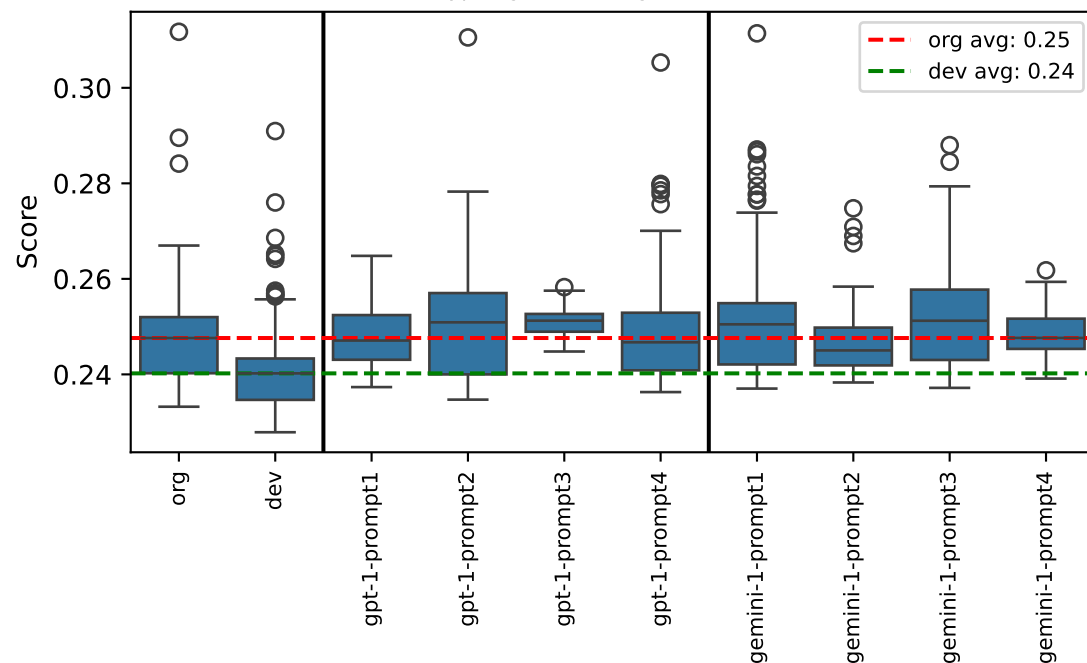


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

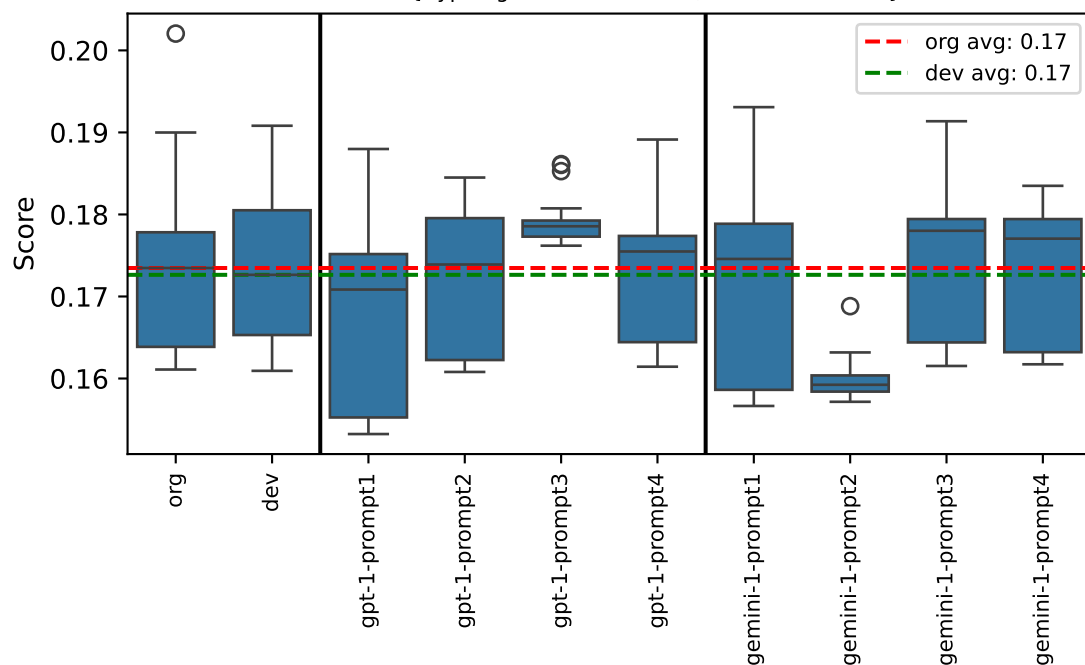
Params: {'typeSignature': 'bigint', 'withNulls': 'true'}



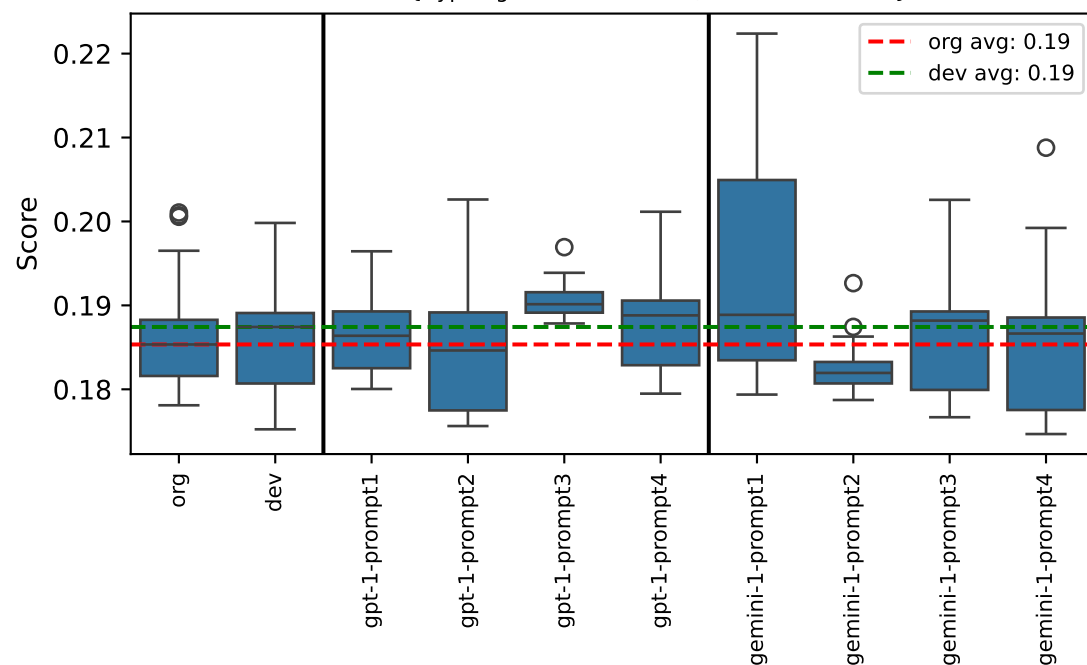
Params: {'typeSignature': 'bigint', 'withNulls': 'false'}



Params: {'typeSignature': 'smallint', 'withNulls': 'true'}

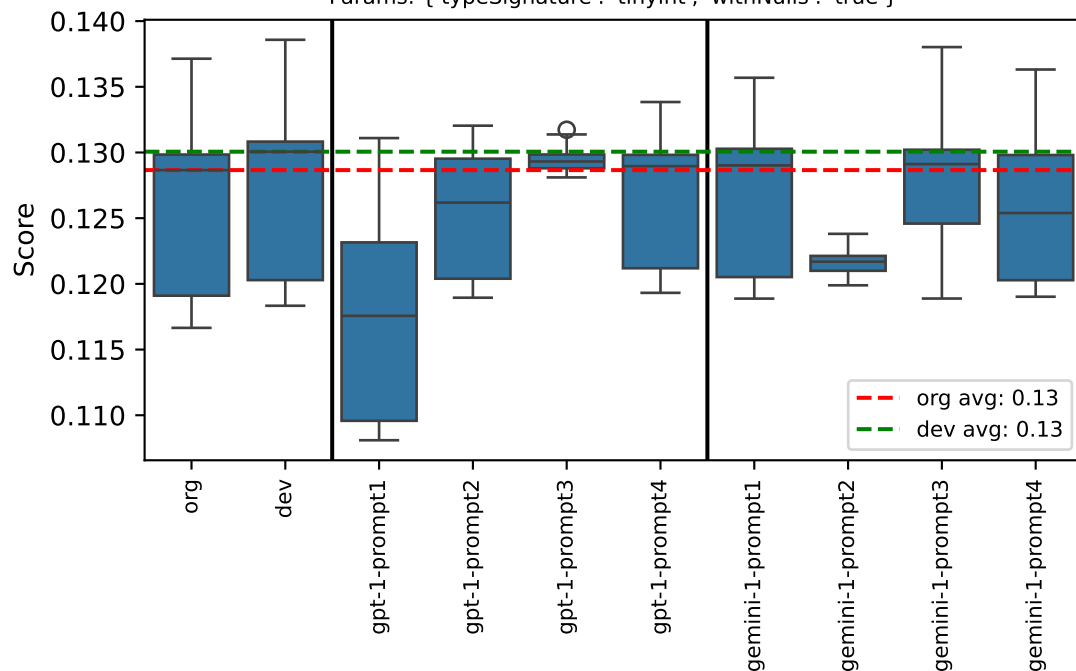


Params: {'typeSignature': 'smallint', 'withNulls': 'false'}

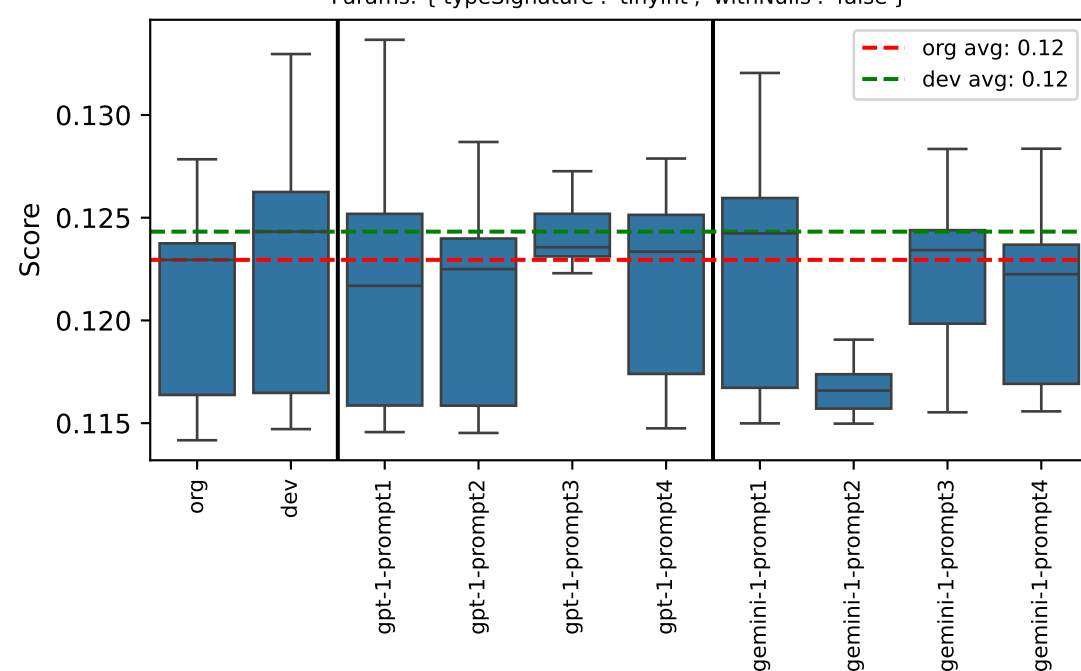


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

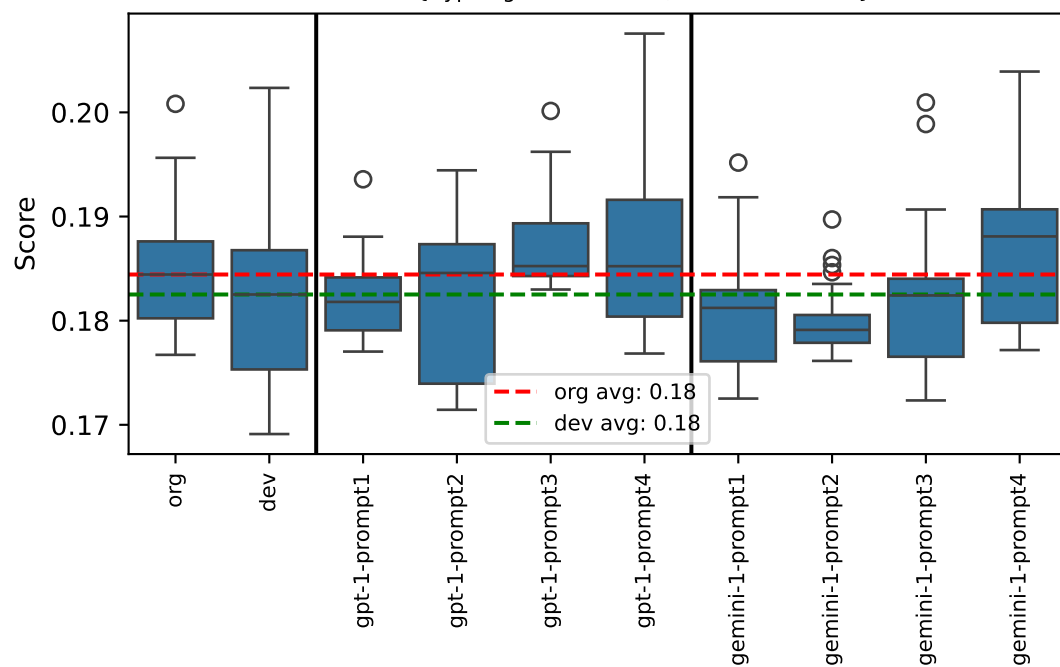
Params: {'typeSignature': 'tinyint', 'withNulls': 'true'}



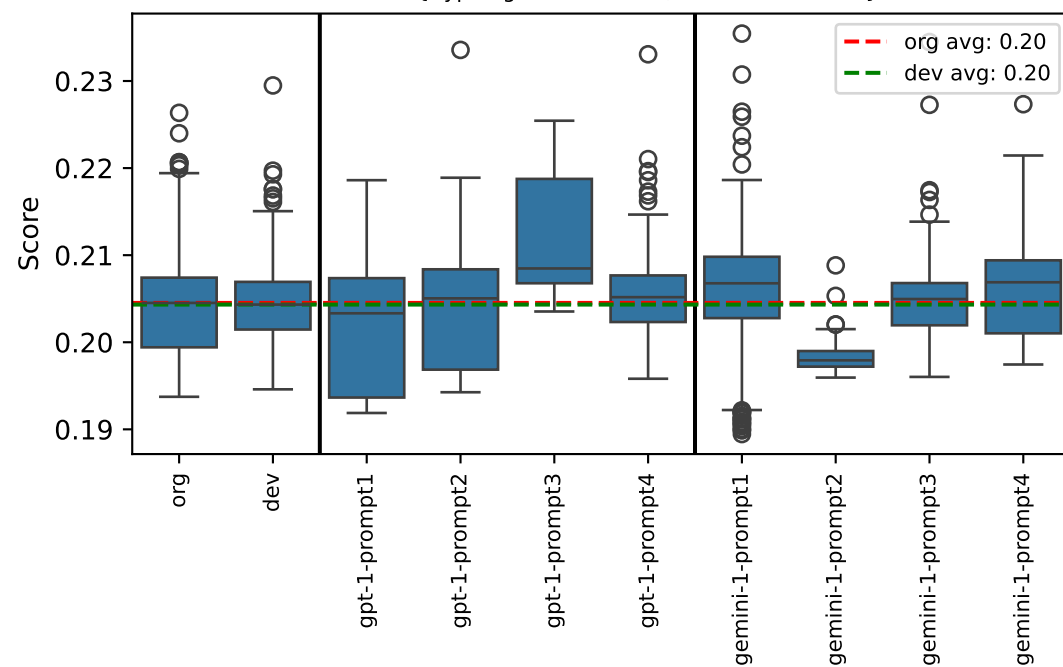
Params: {'typeSignature': 'tinyint', 'withNulls': 'false'}



Params: {'typeSignature': 'date', 'withNulls': 'true'}

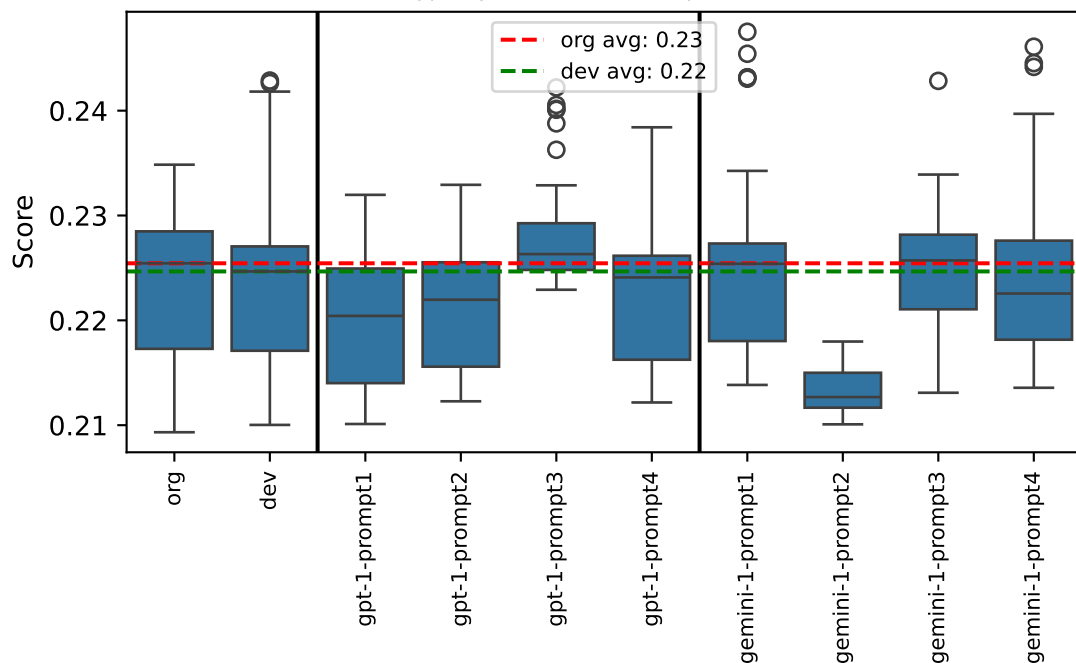


Params: {'typeSignature': 'date', 'withNulls': 'false'}

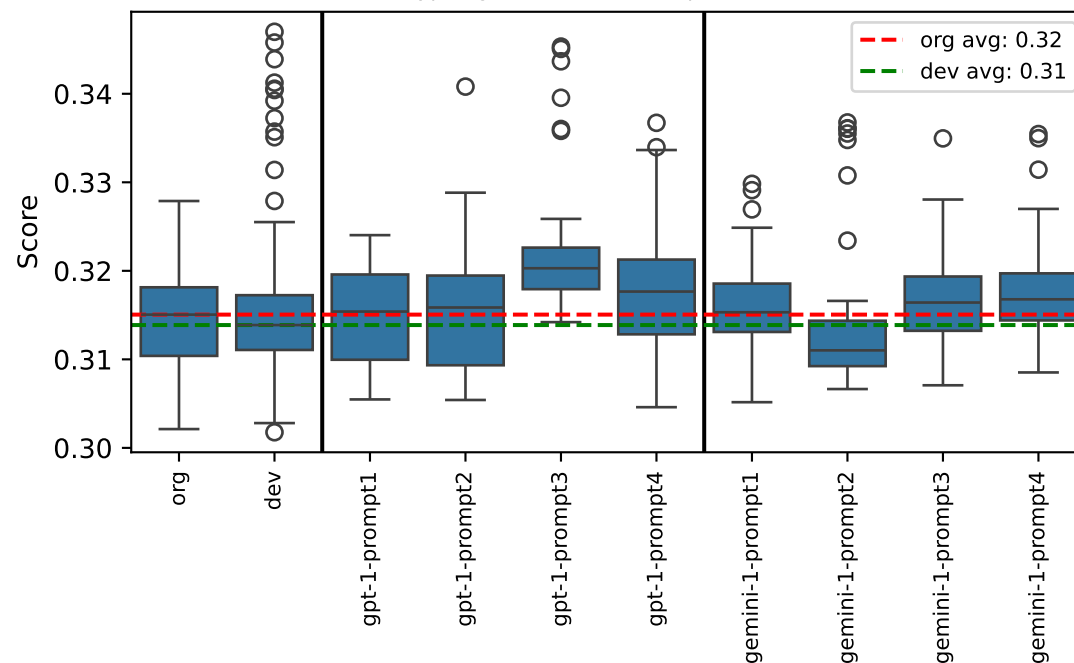


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

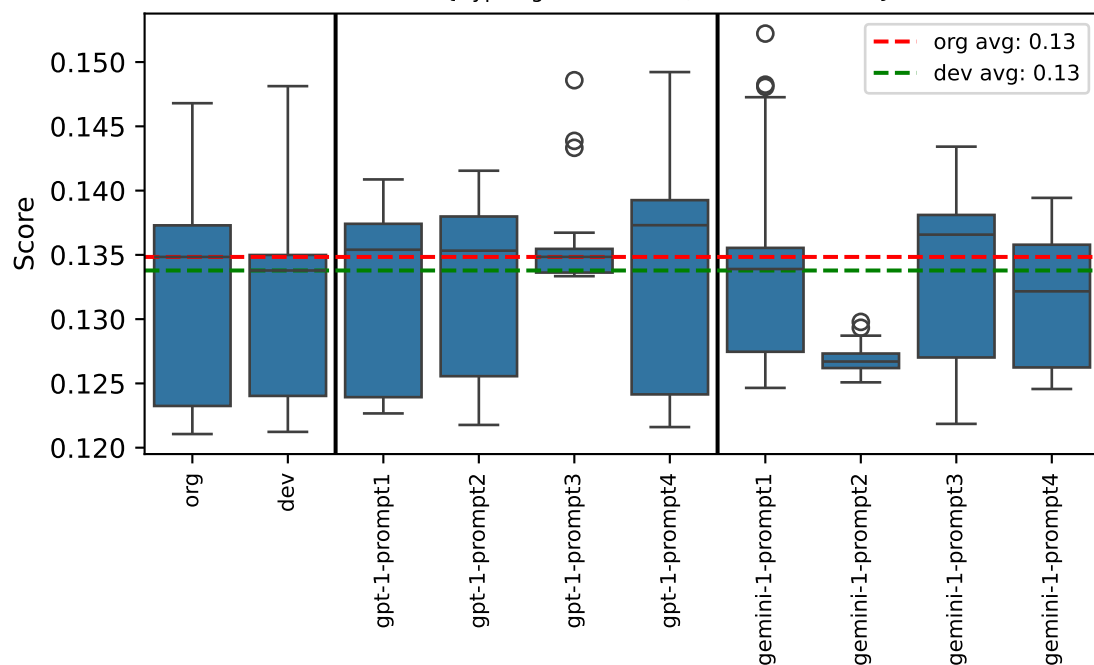
Params: {'typeSignature': 'timestamp', 'withNulls': 'true'}



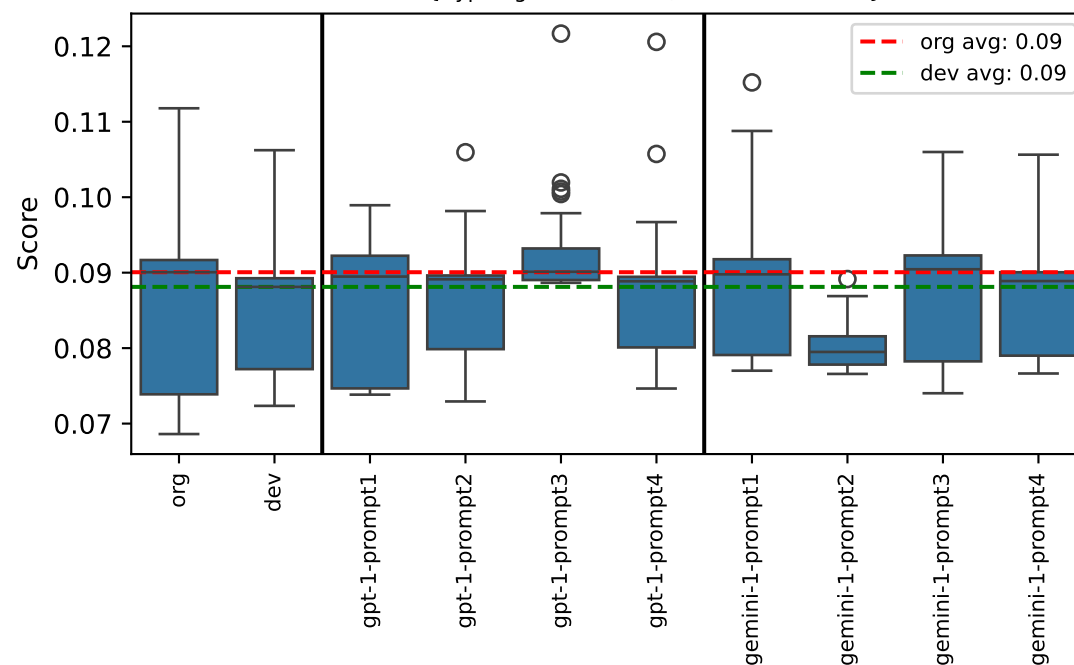
Params: {'typeSignature': 'timestamp', 'withNulls': 'false'}



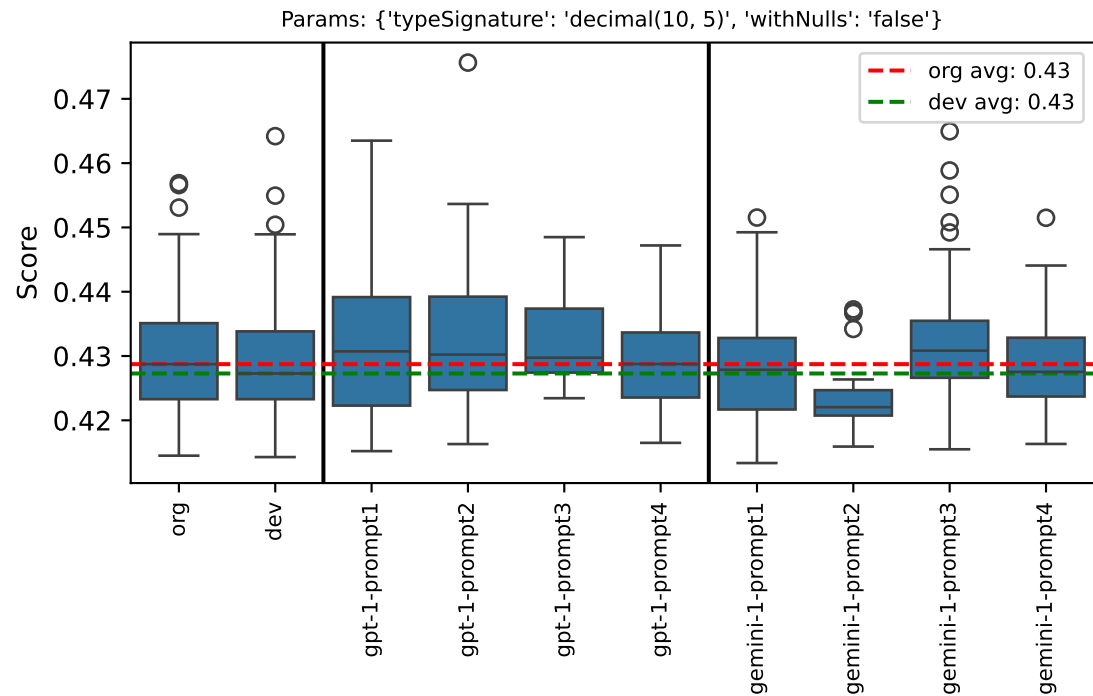
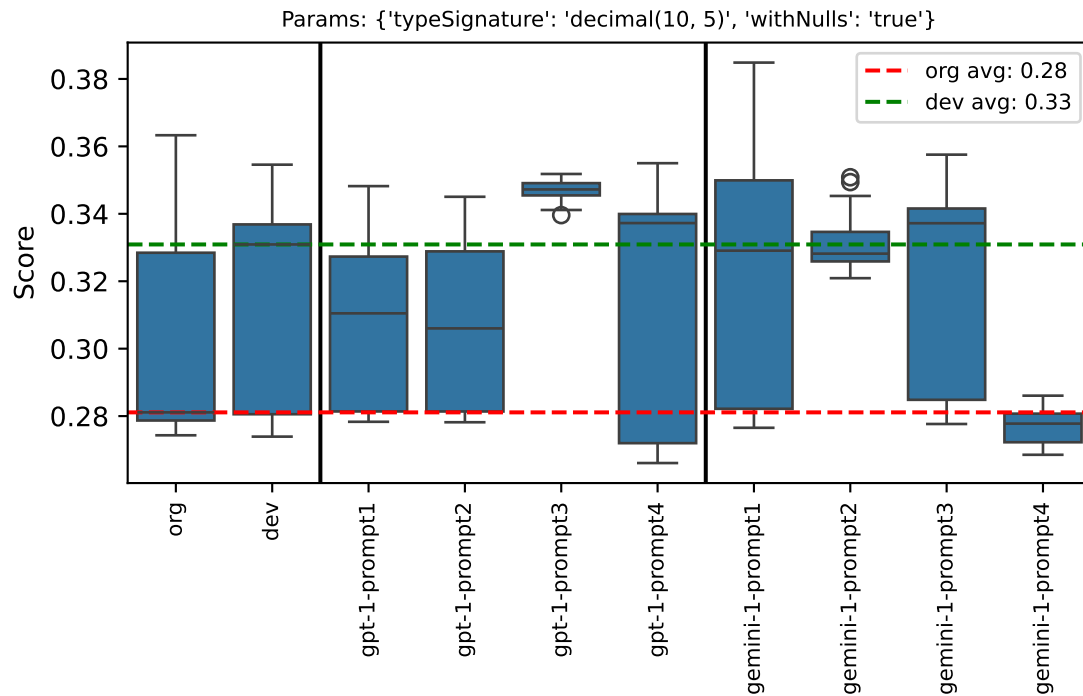
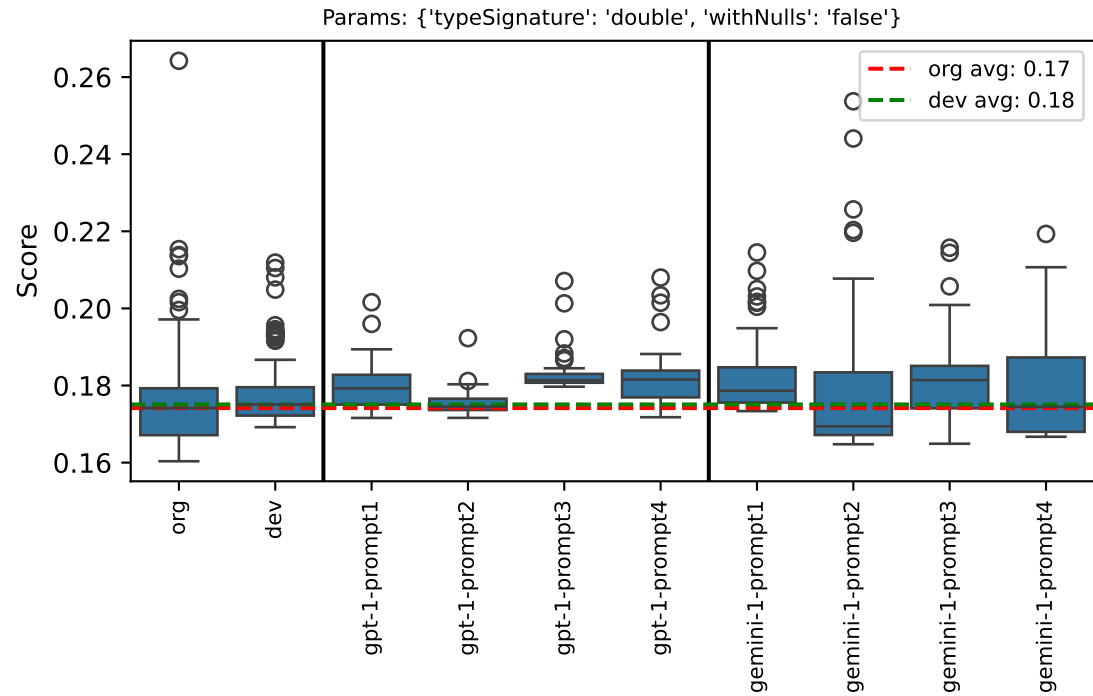
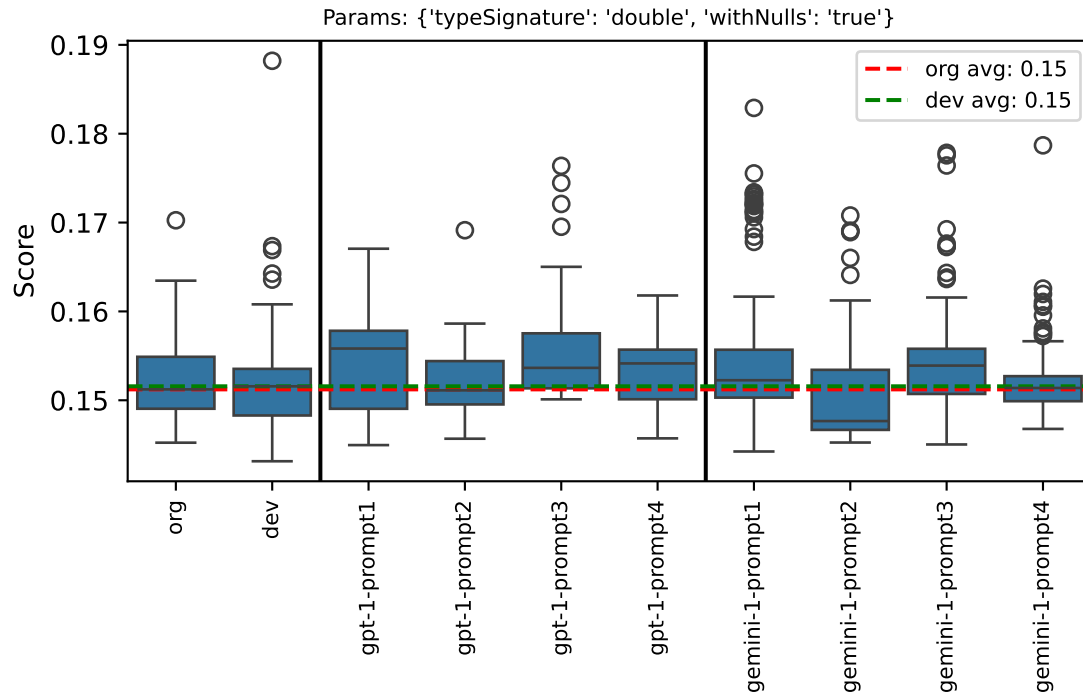
Params: {'typeSignature': 'real', 'withNulls': 'true'}



Params: {'typeSignature': 'real', 'withNulls': 'false'}

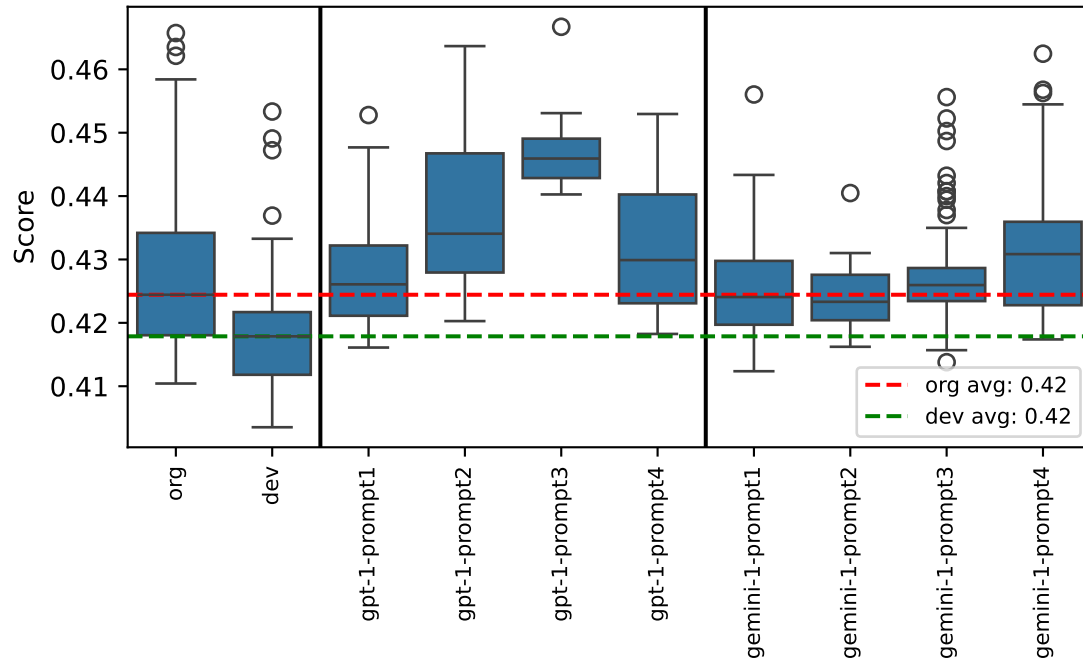


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

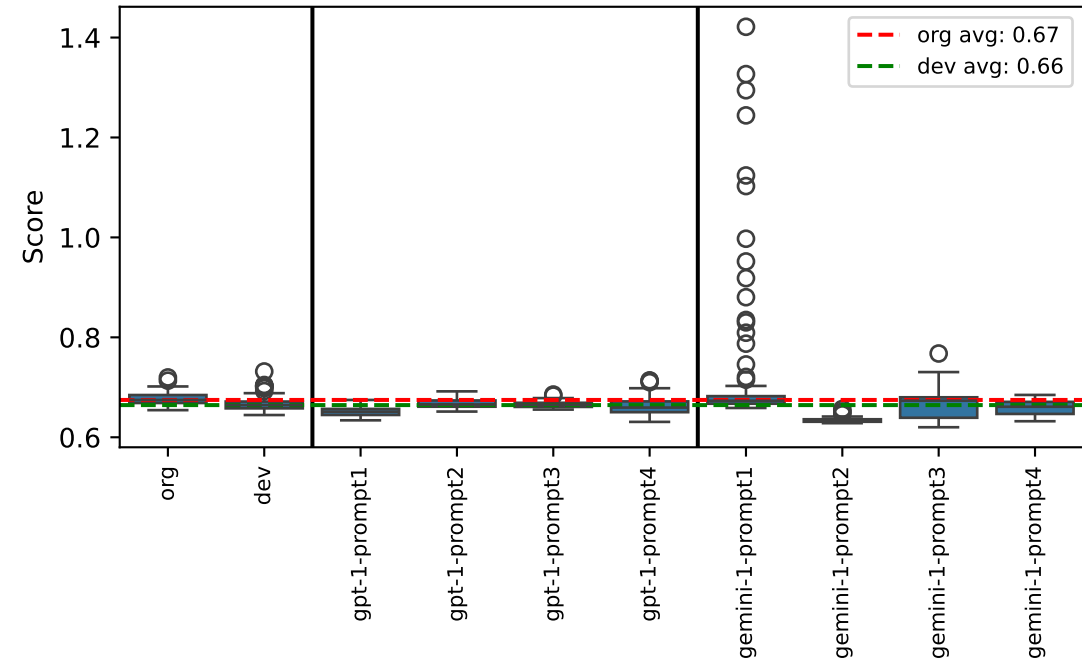


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

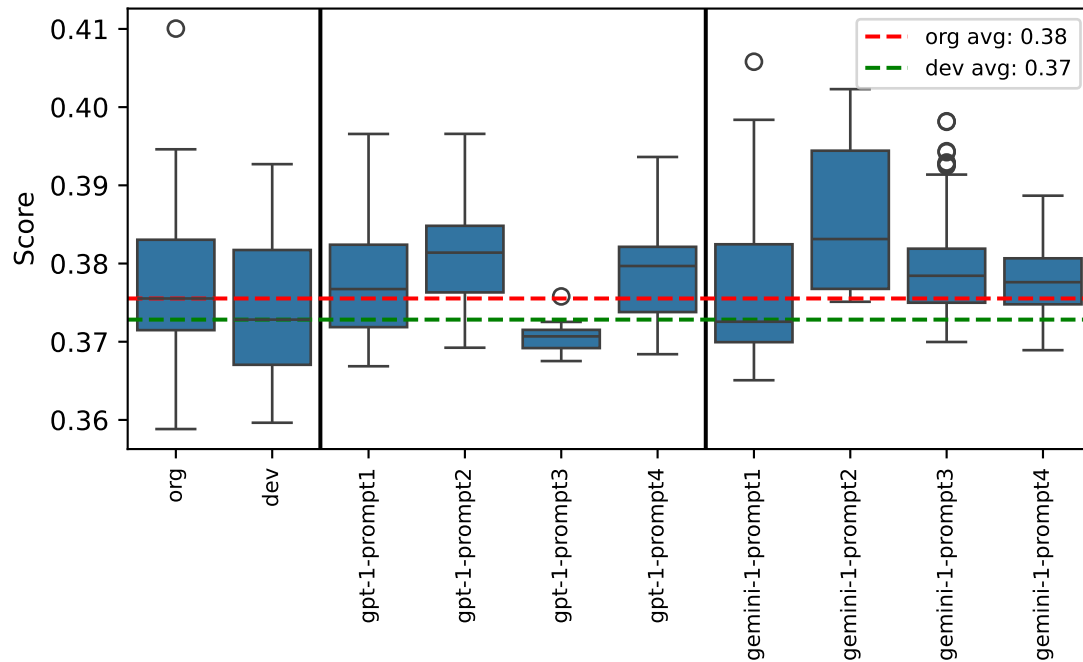
Params: {'typeSignature': 'decimal(30, 10)', 'withNulls': 'true'}



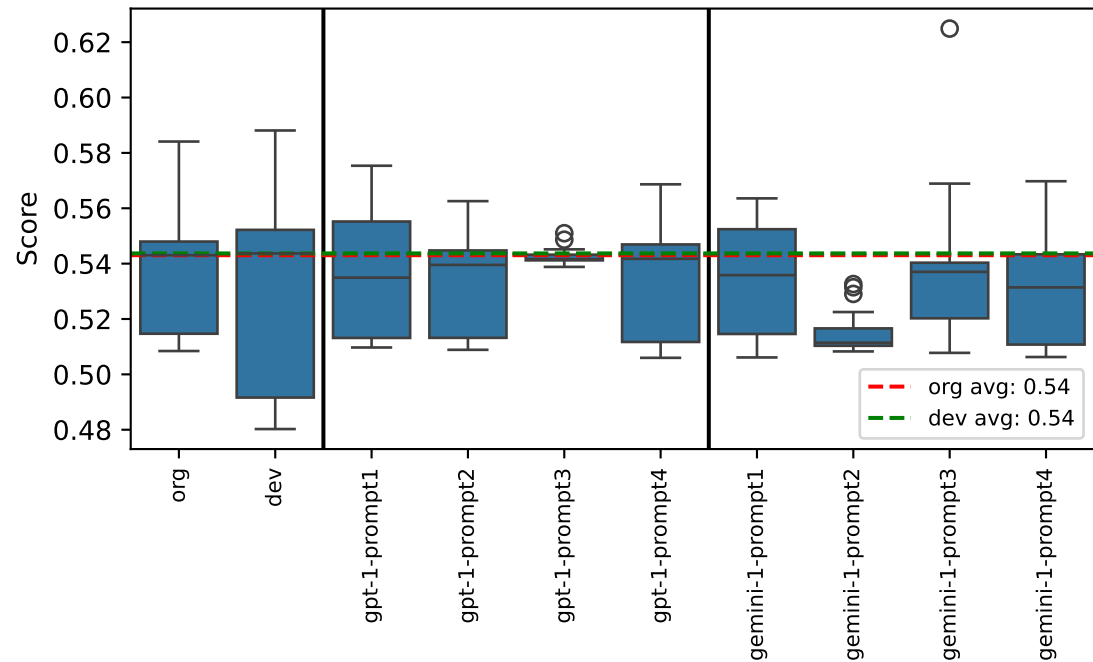
Params: {'typeSignature': 'decimal(30, 10)', 'withNulls': 'false'}



Params: {'typeSignature': 'varchar_direct', 'withNulls': 'true'}

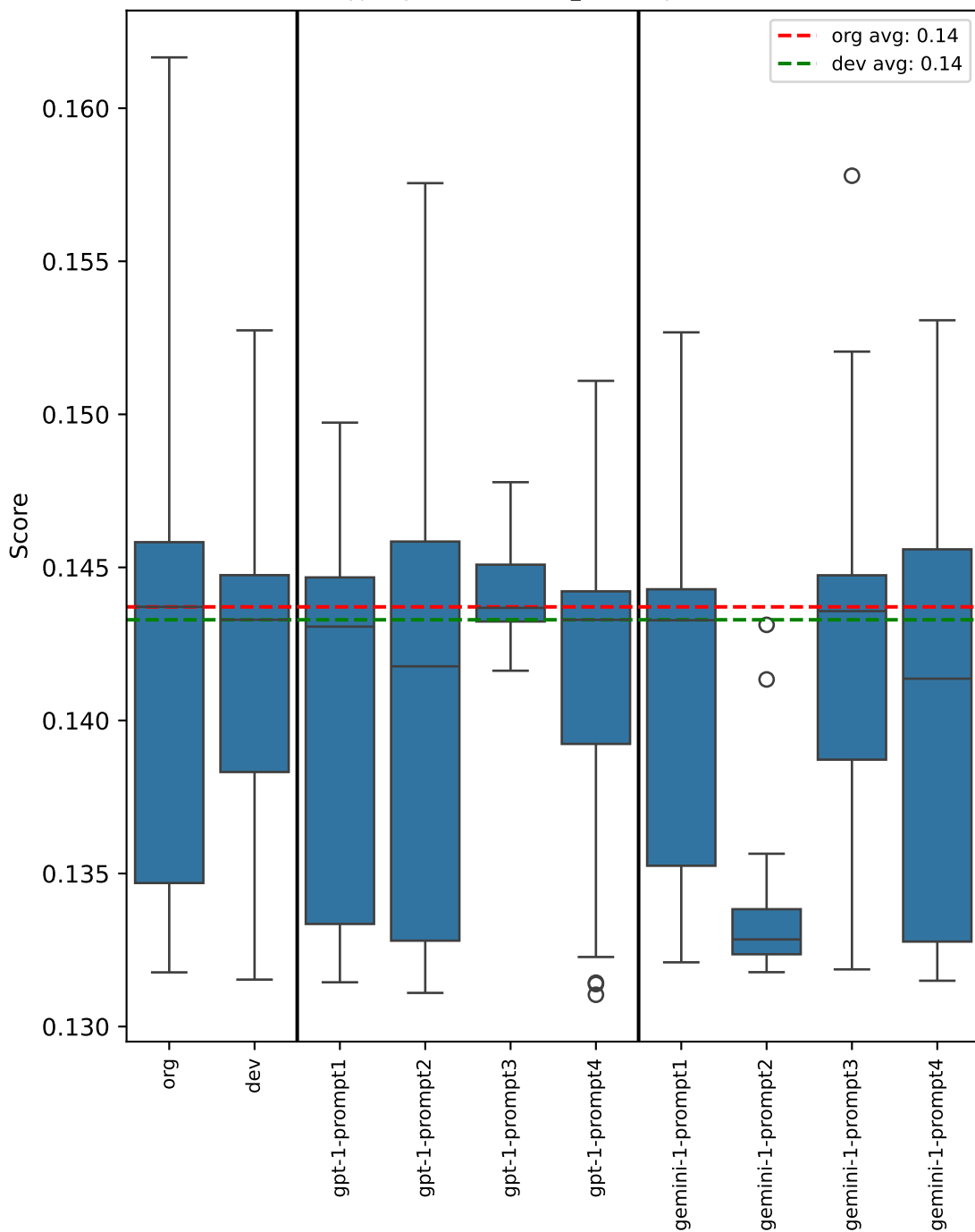


Params: {'typeSignature': 'varchar_direct', 'withNulls': 'false'}

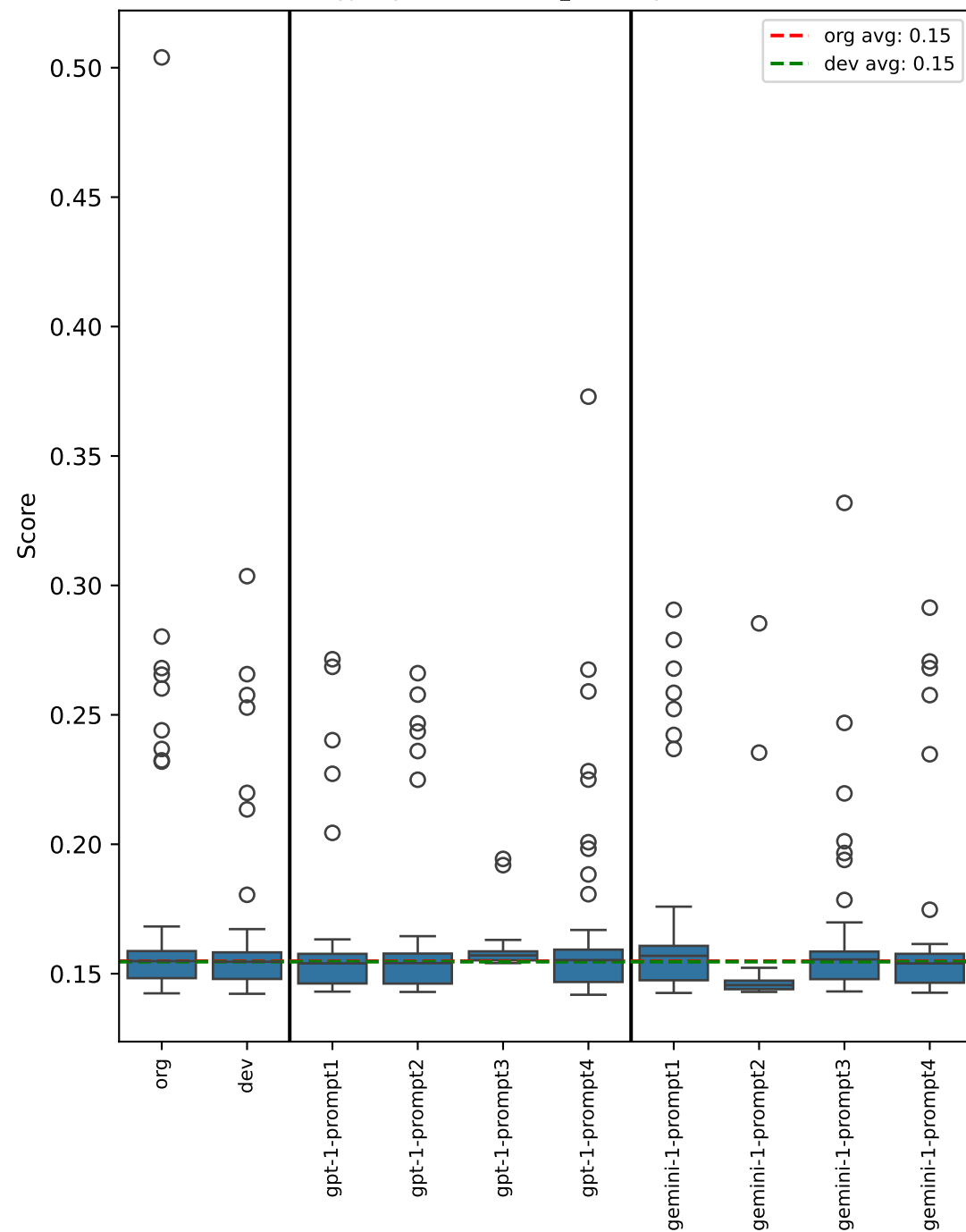


Benchmark: com.facebook.presto.orc.BenchmarkSelectiveStreamReaders.readWithFilter

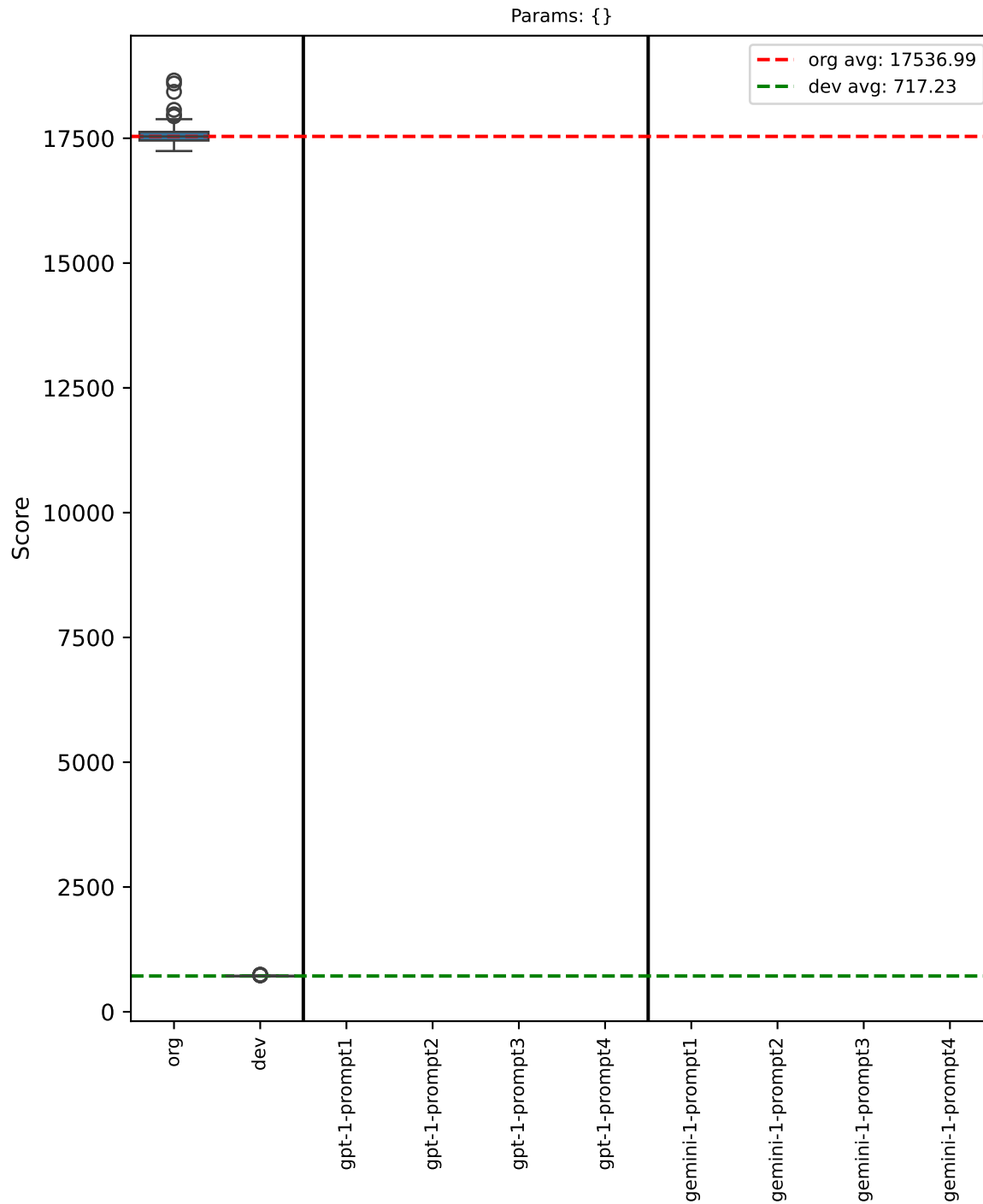
Params: {'typeSignature': 'varchar_dictionary', 'withNulls': 'true'}



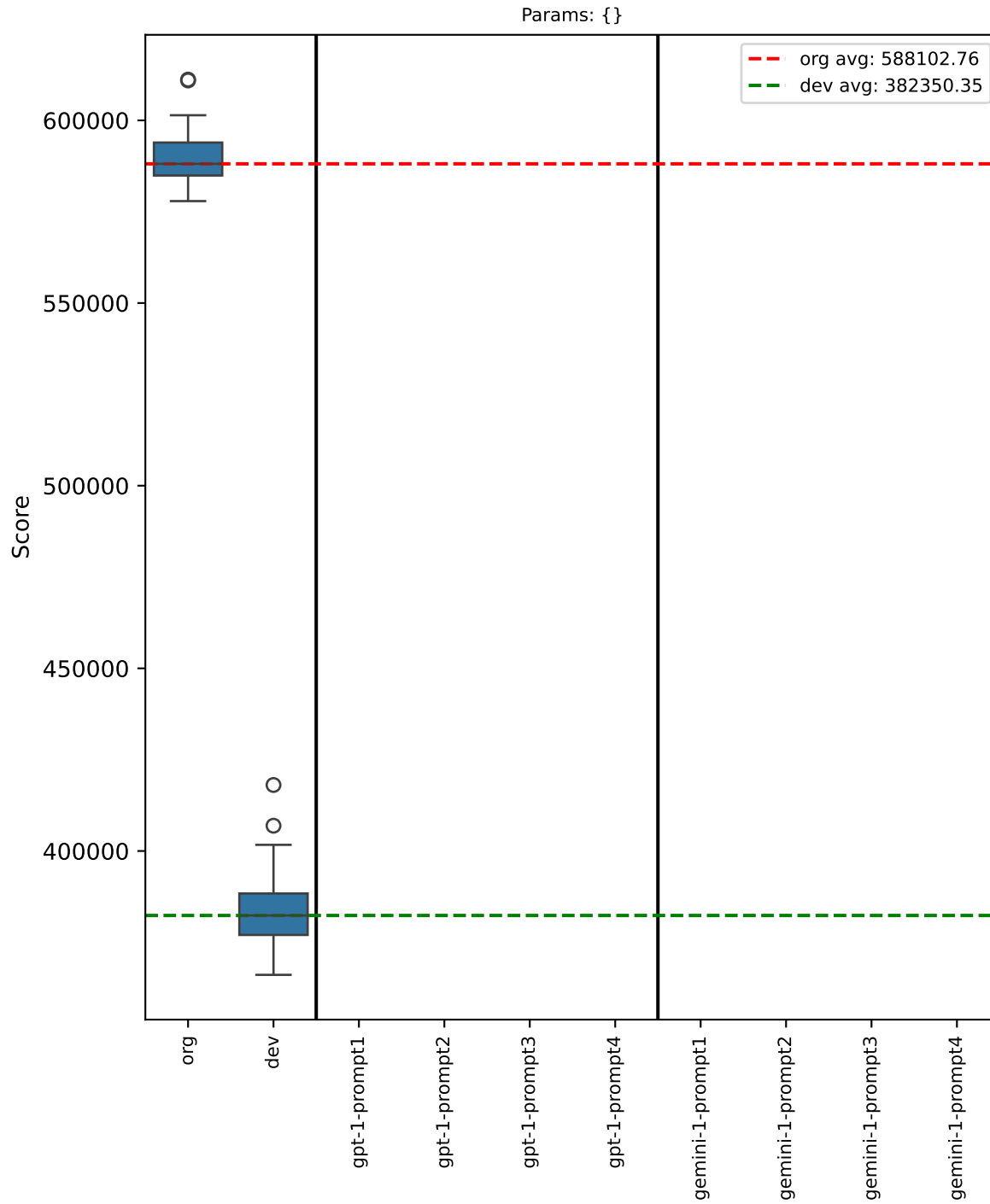
Params: {'typeSignature': 'varchar_dictionary', 'withNulls': 'false'}



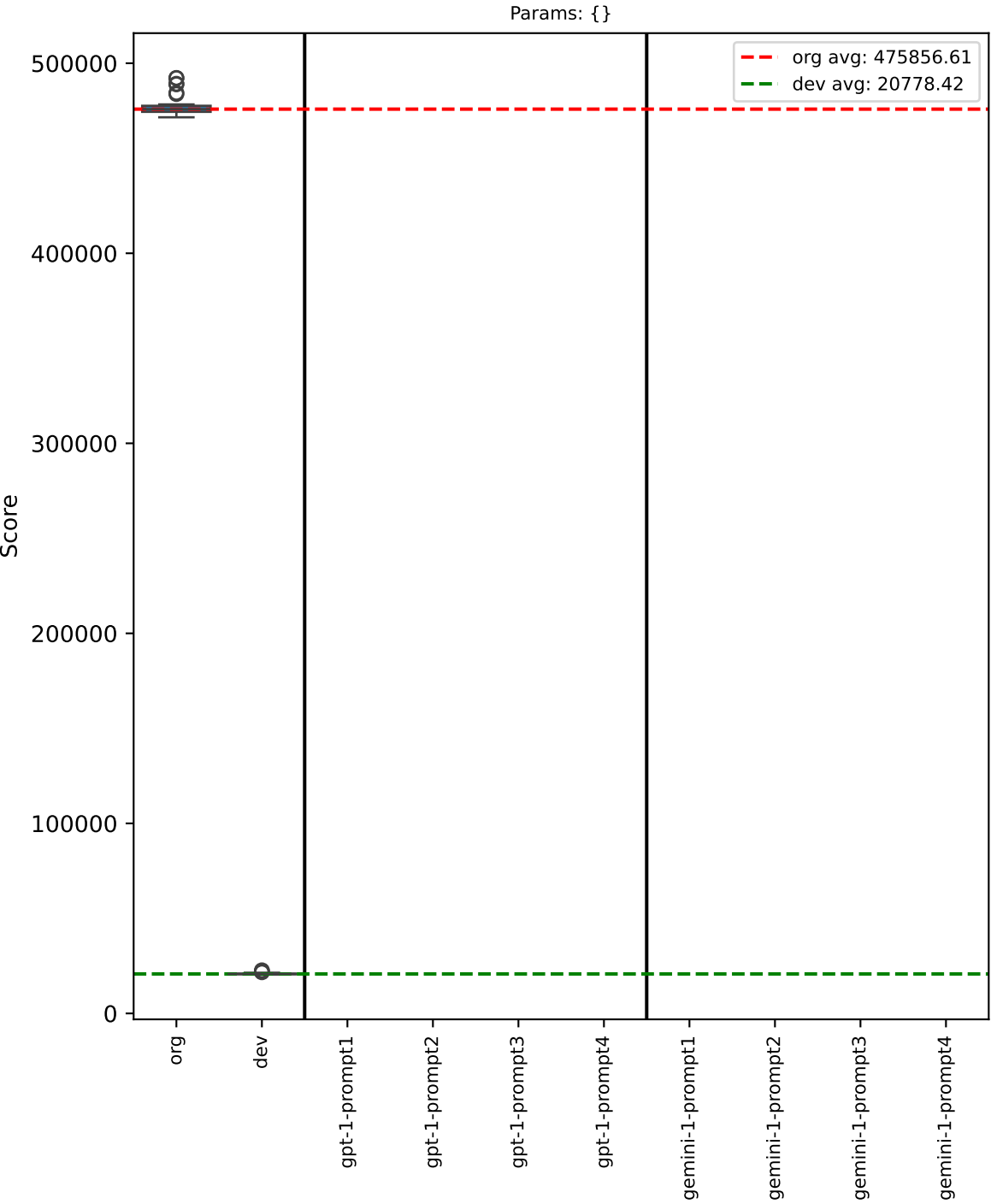
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferLineStringCircle



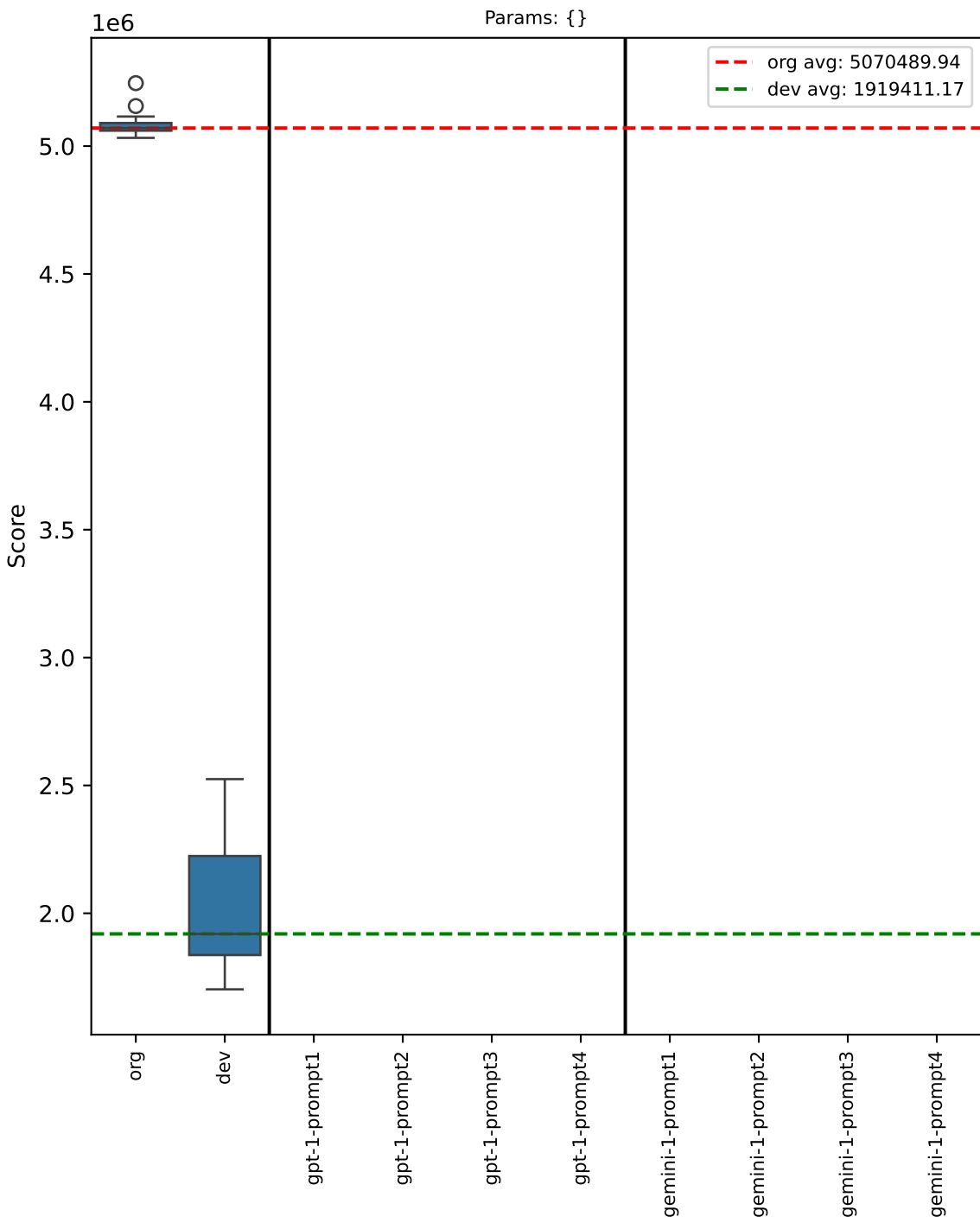
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferLineStringDense



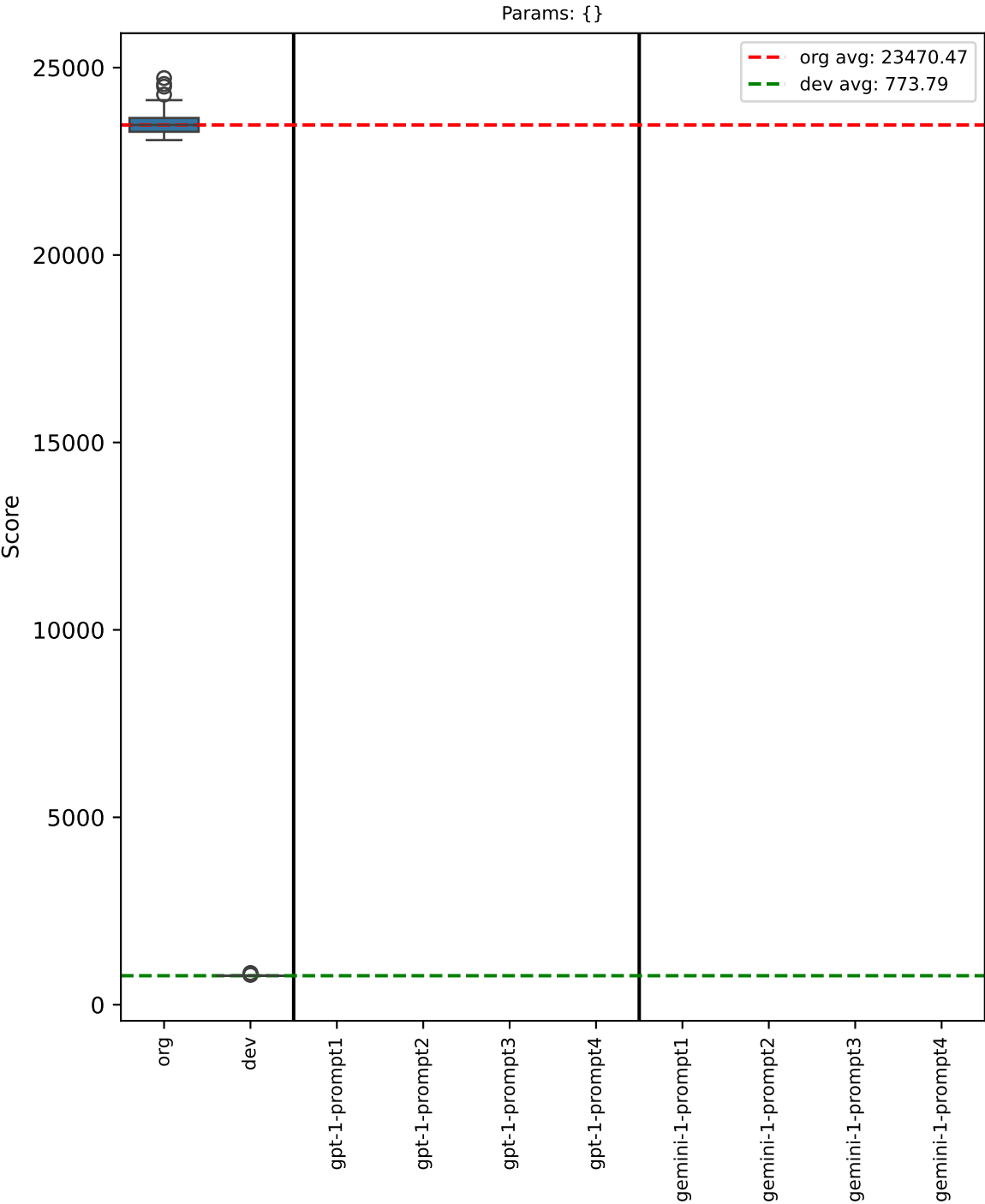
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferMultiPointDense



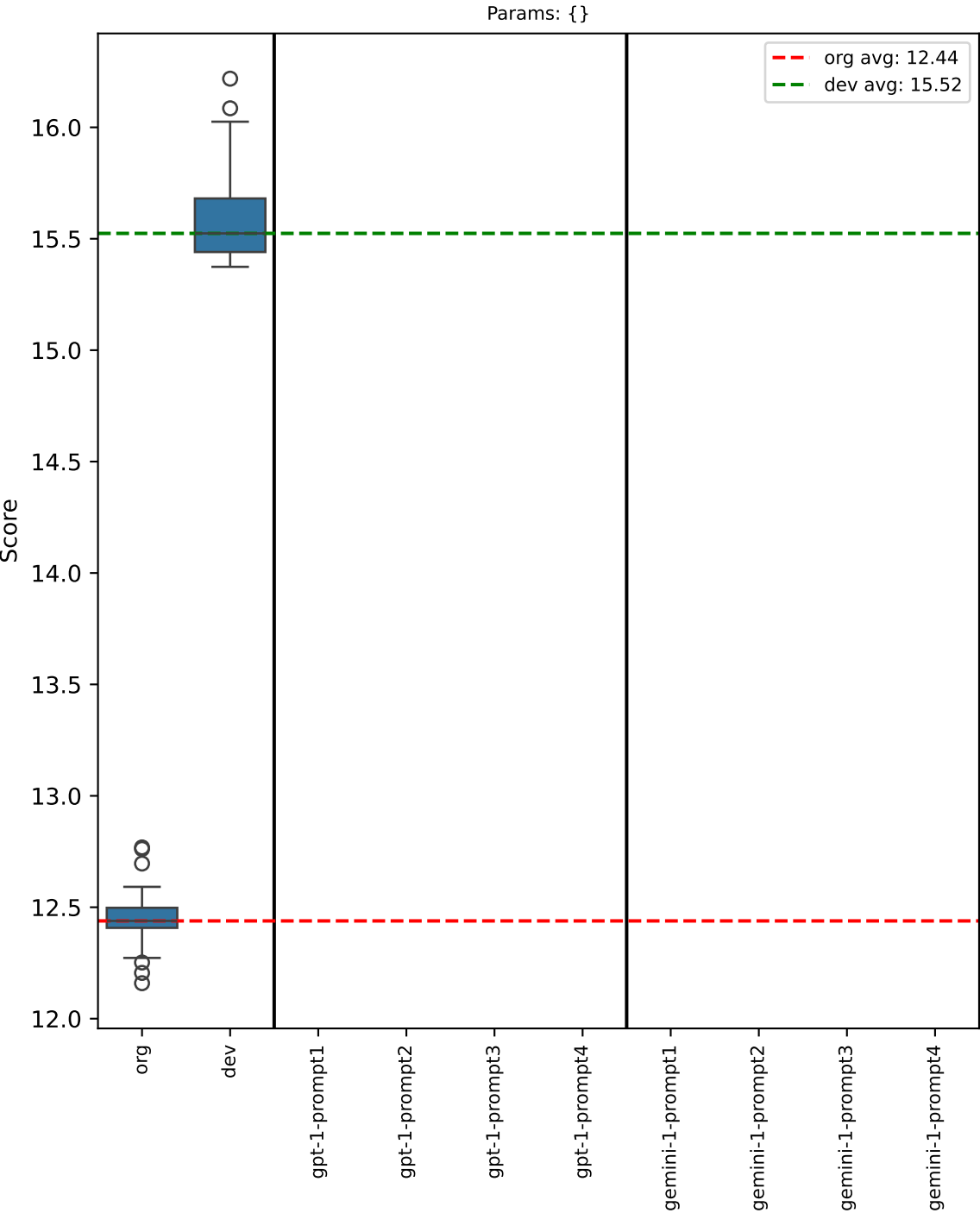
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferMultiPointReallyDense



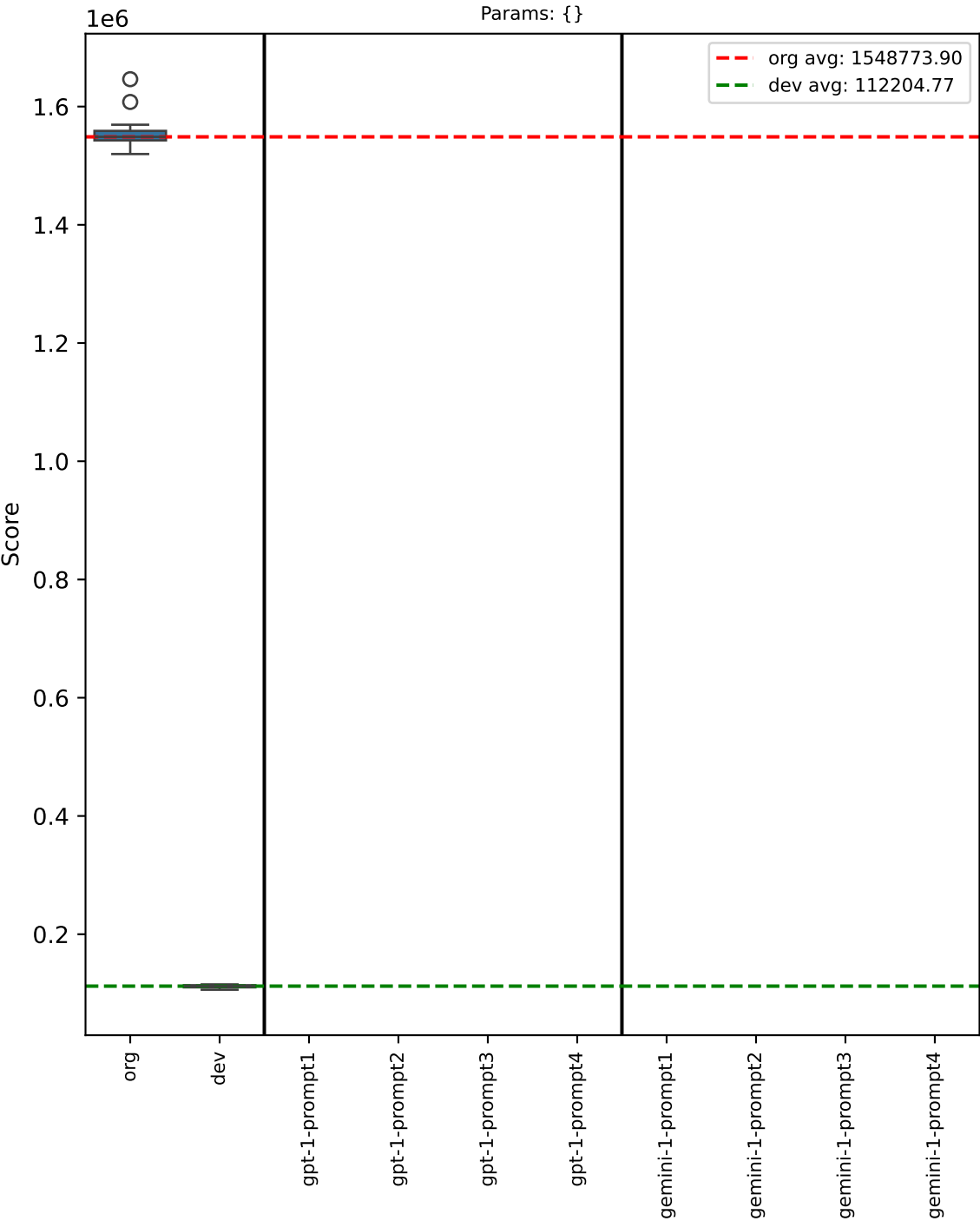
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferMultiPointSparse



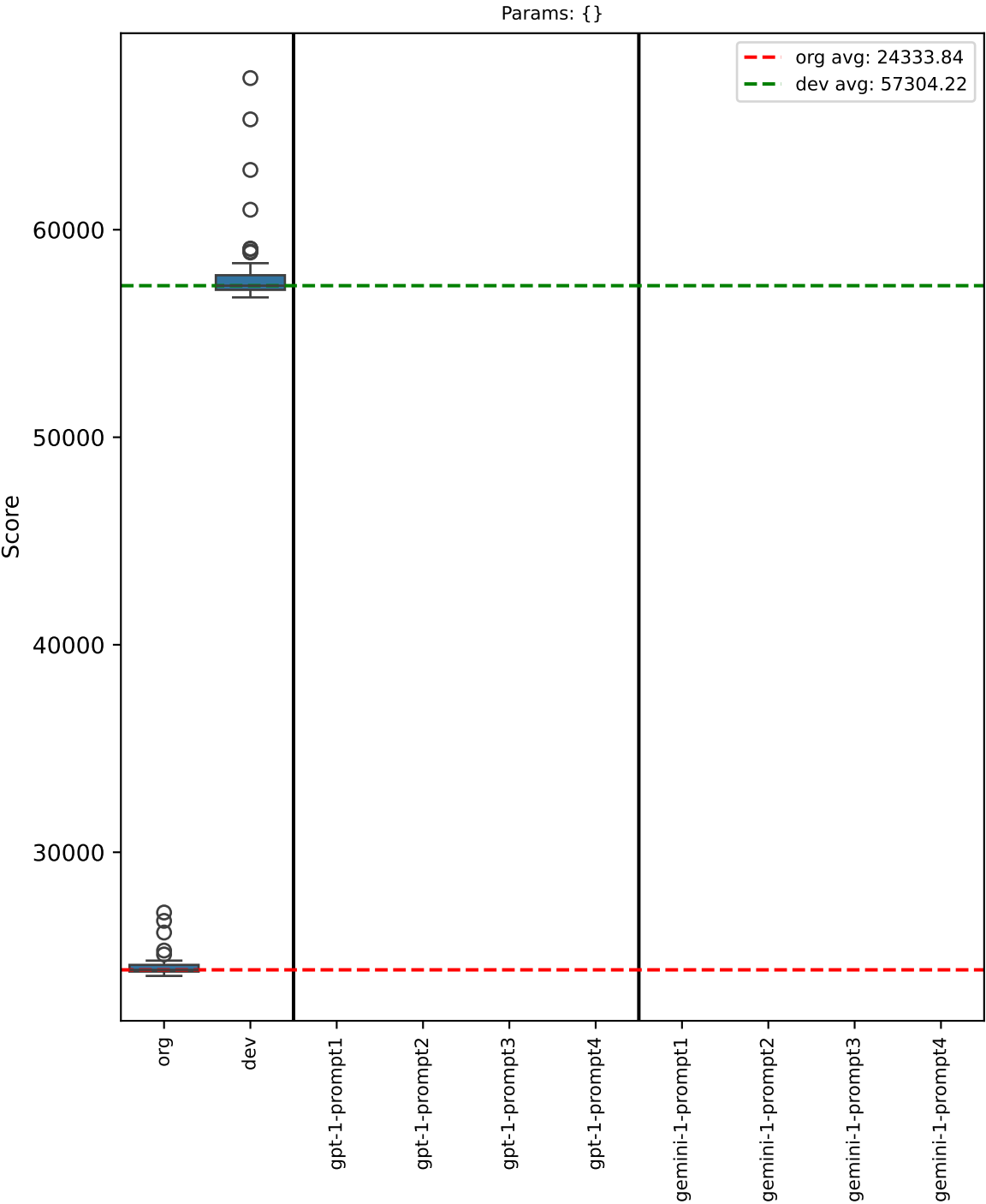
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferPoint



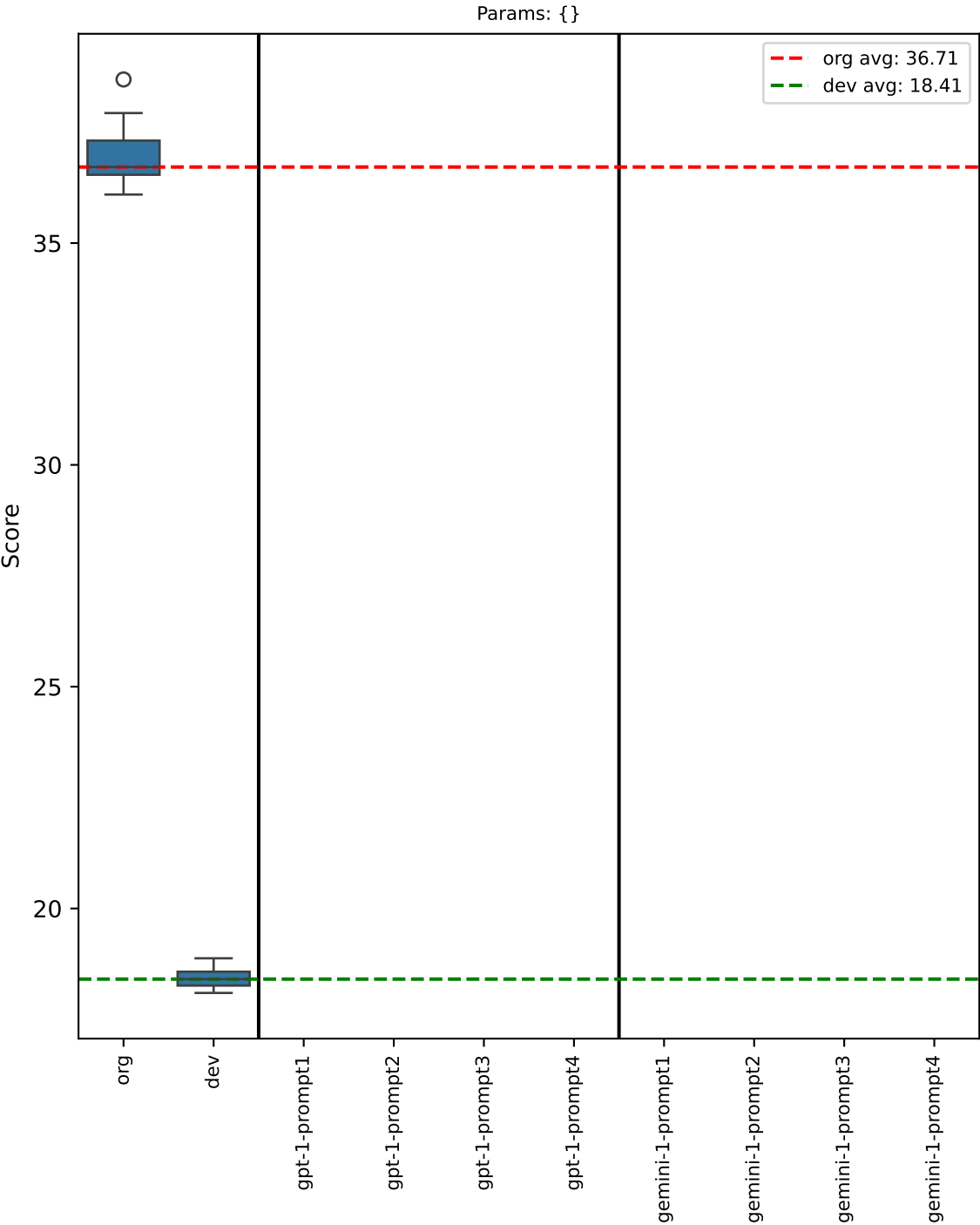
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferPolygonDense



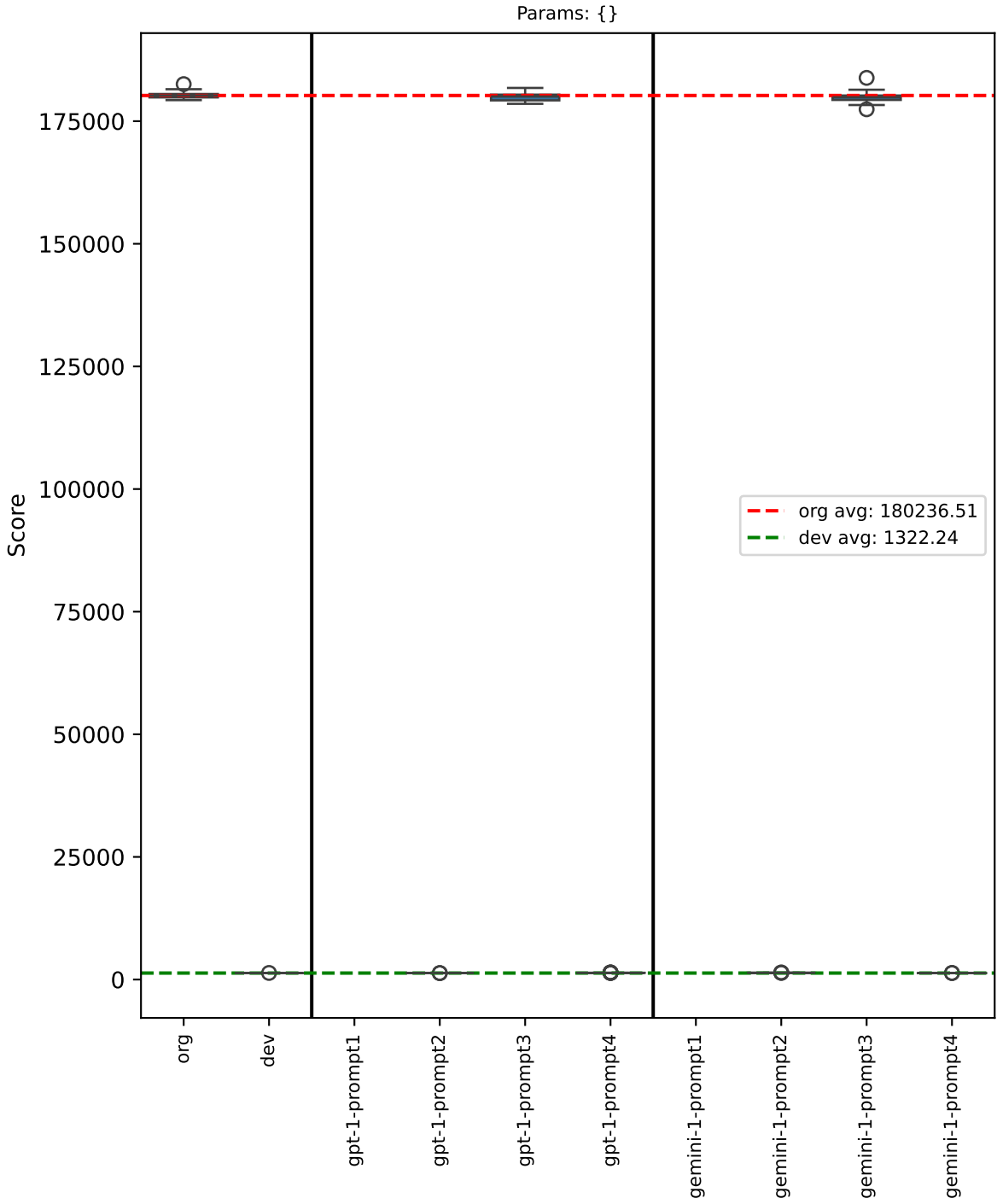
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferPolygonNormal



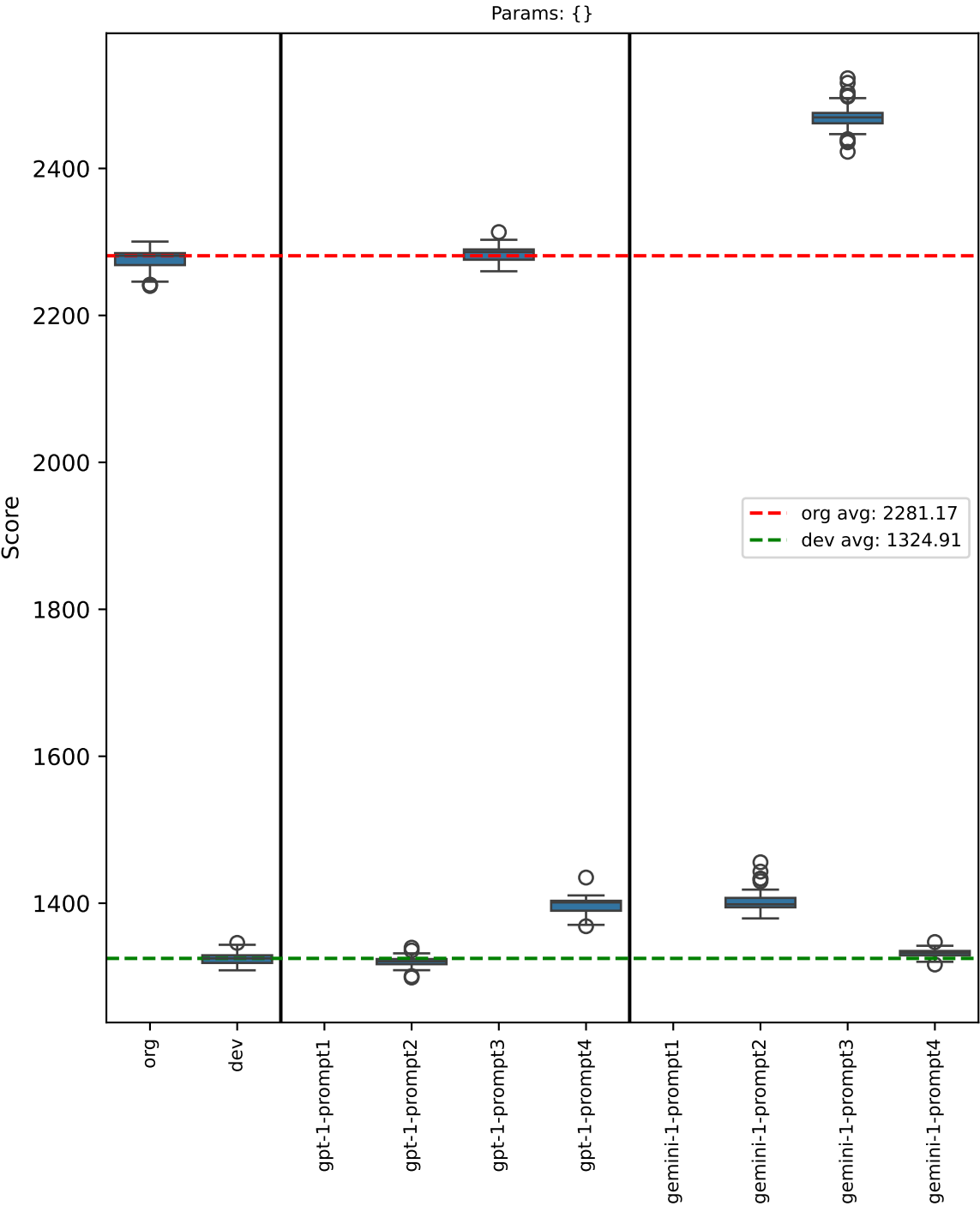
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTBuffer.stBufferPolygonSimple

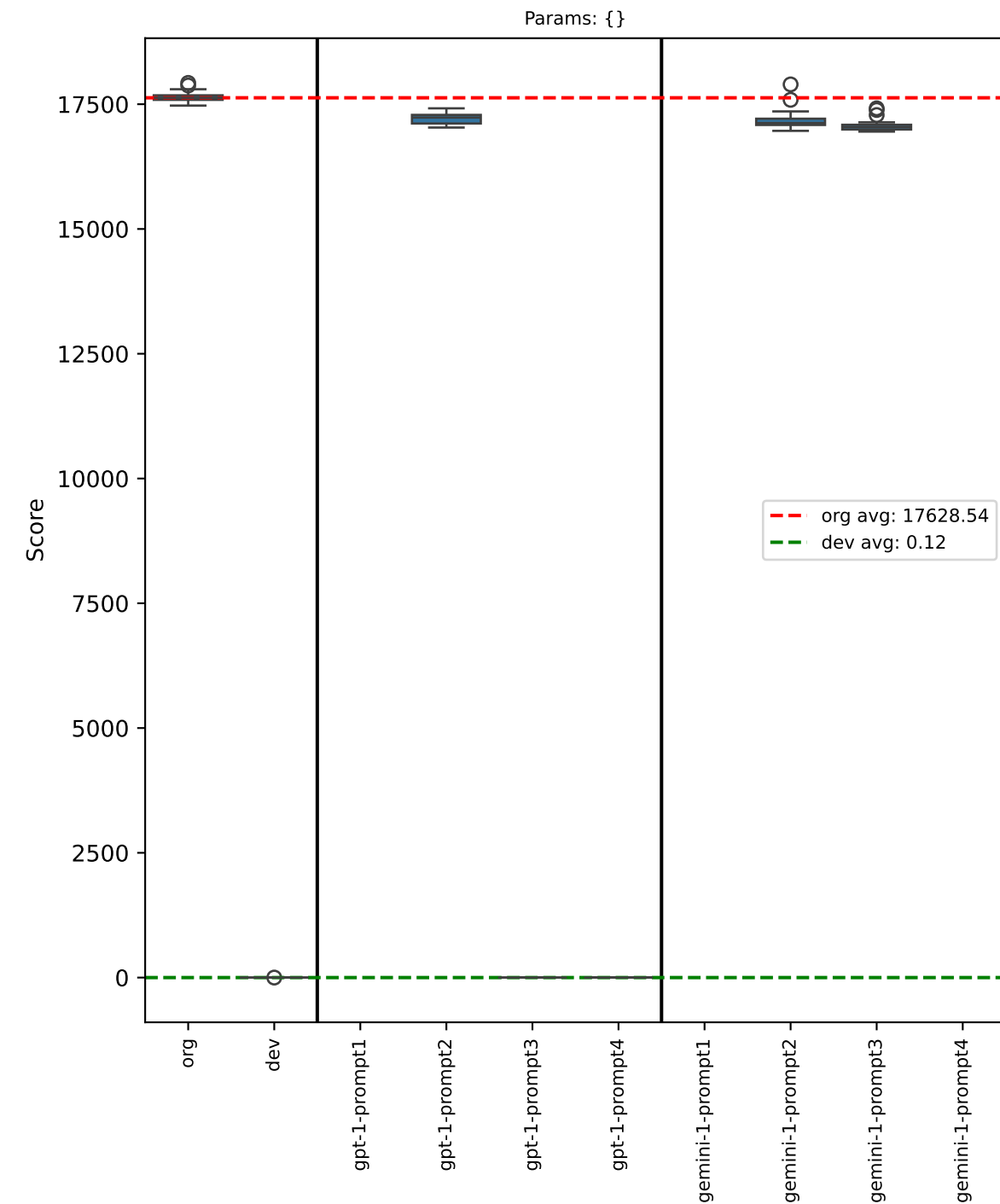


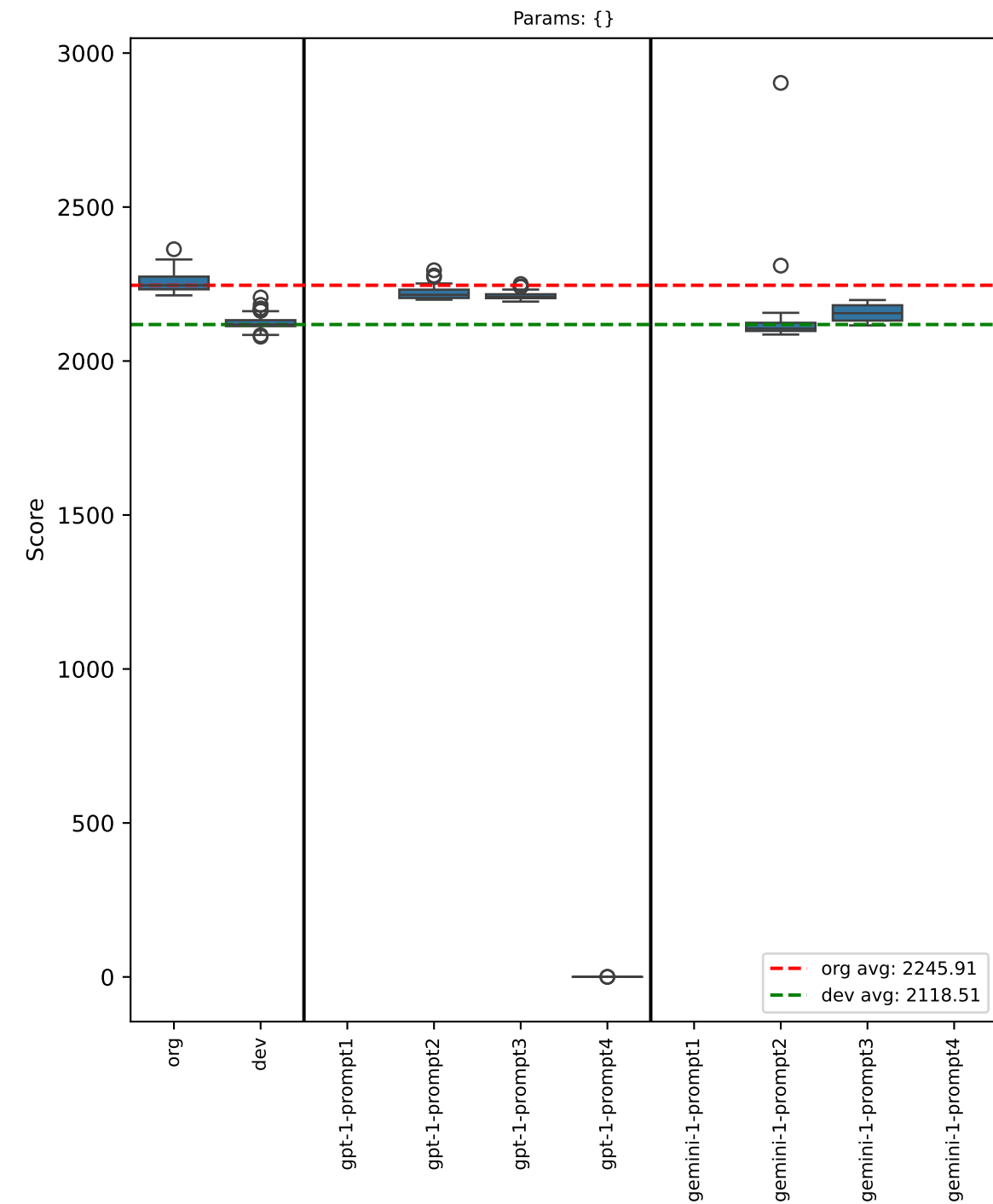
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTEnvelope.complexGeometry

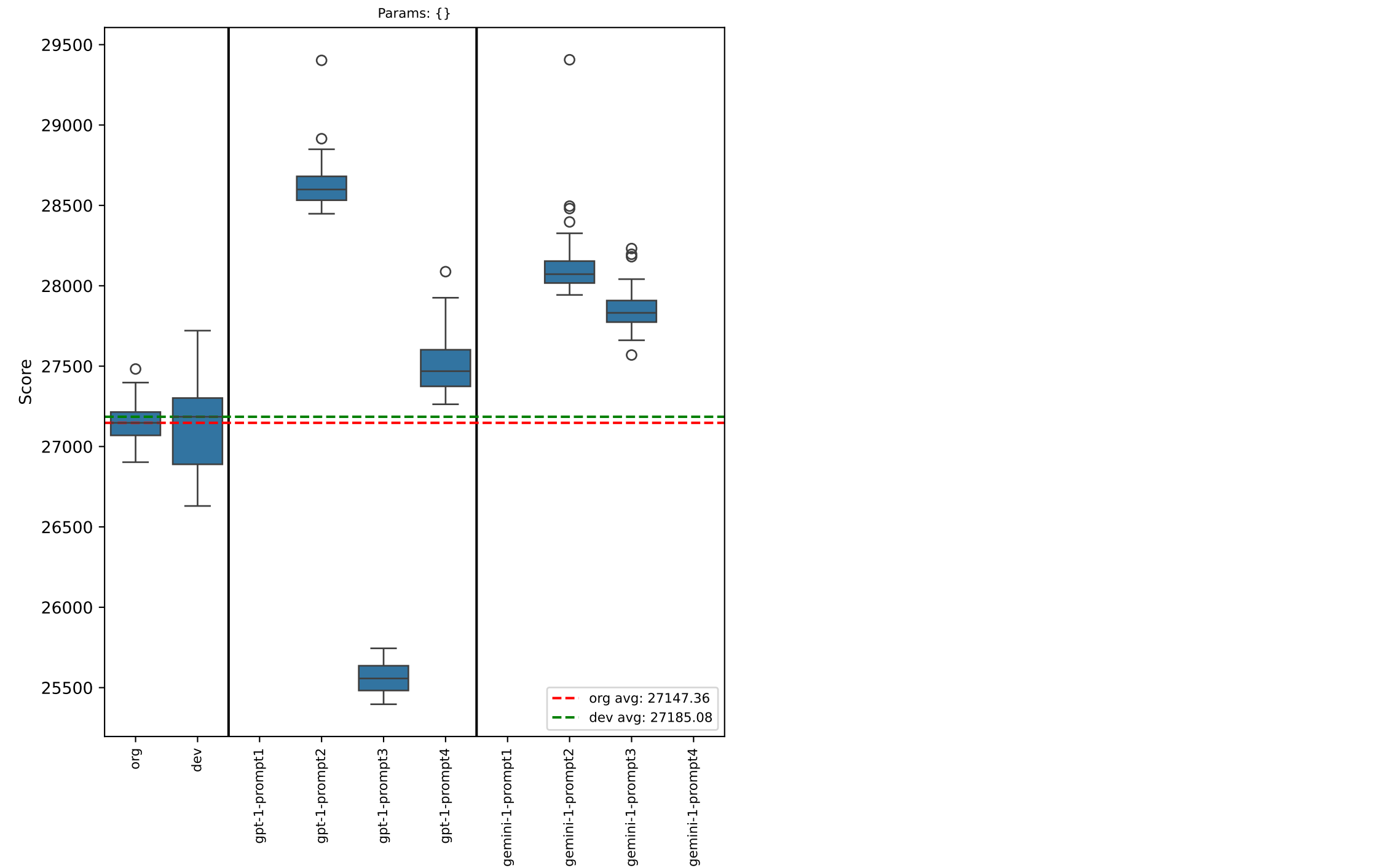


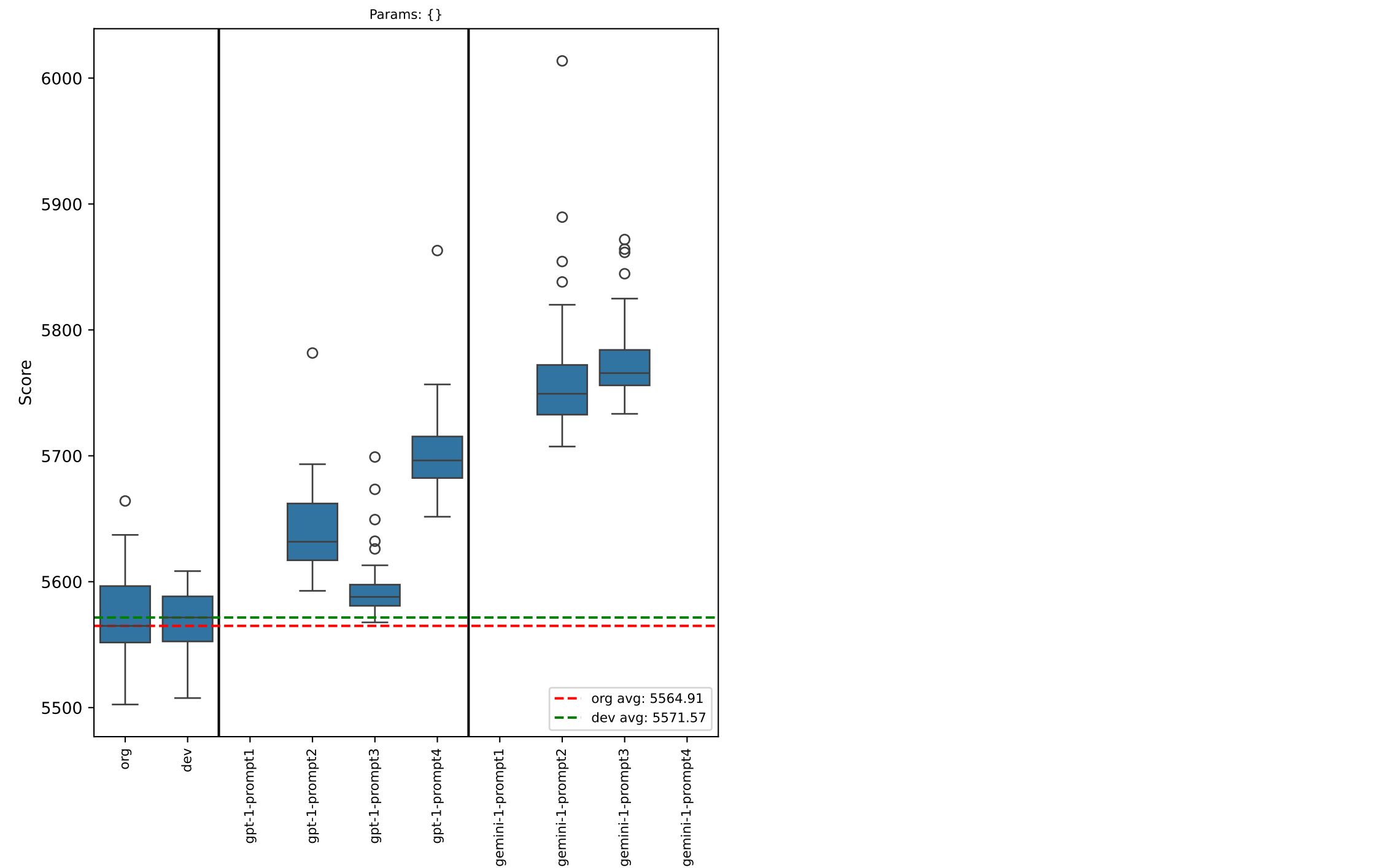
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTEnvelope.simpleGeometry

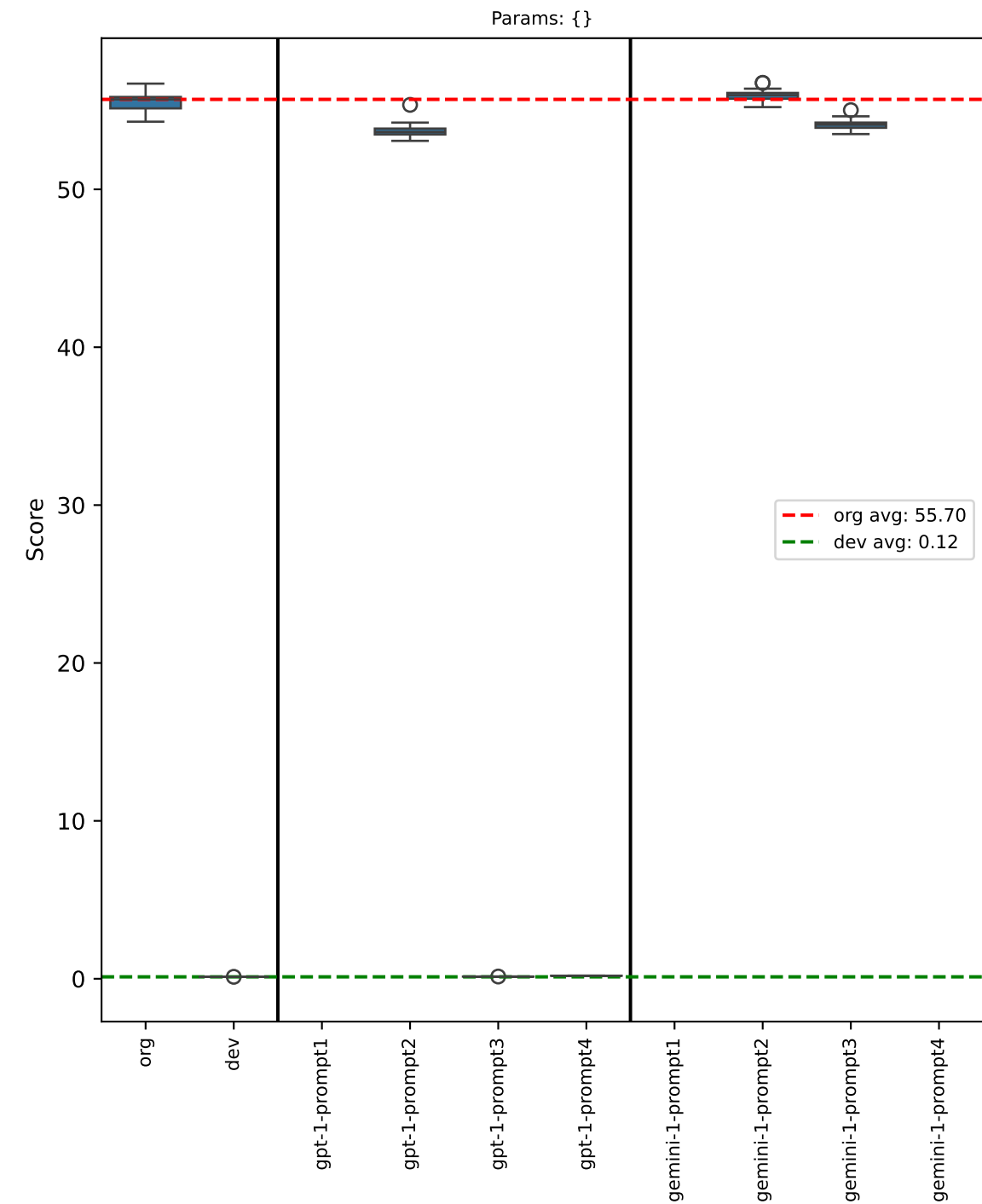


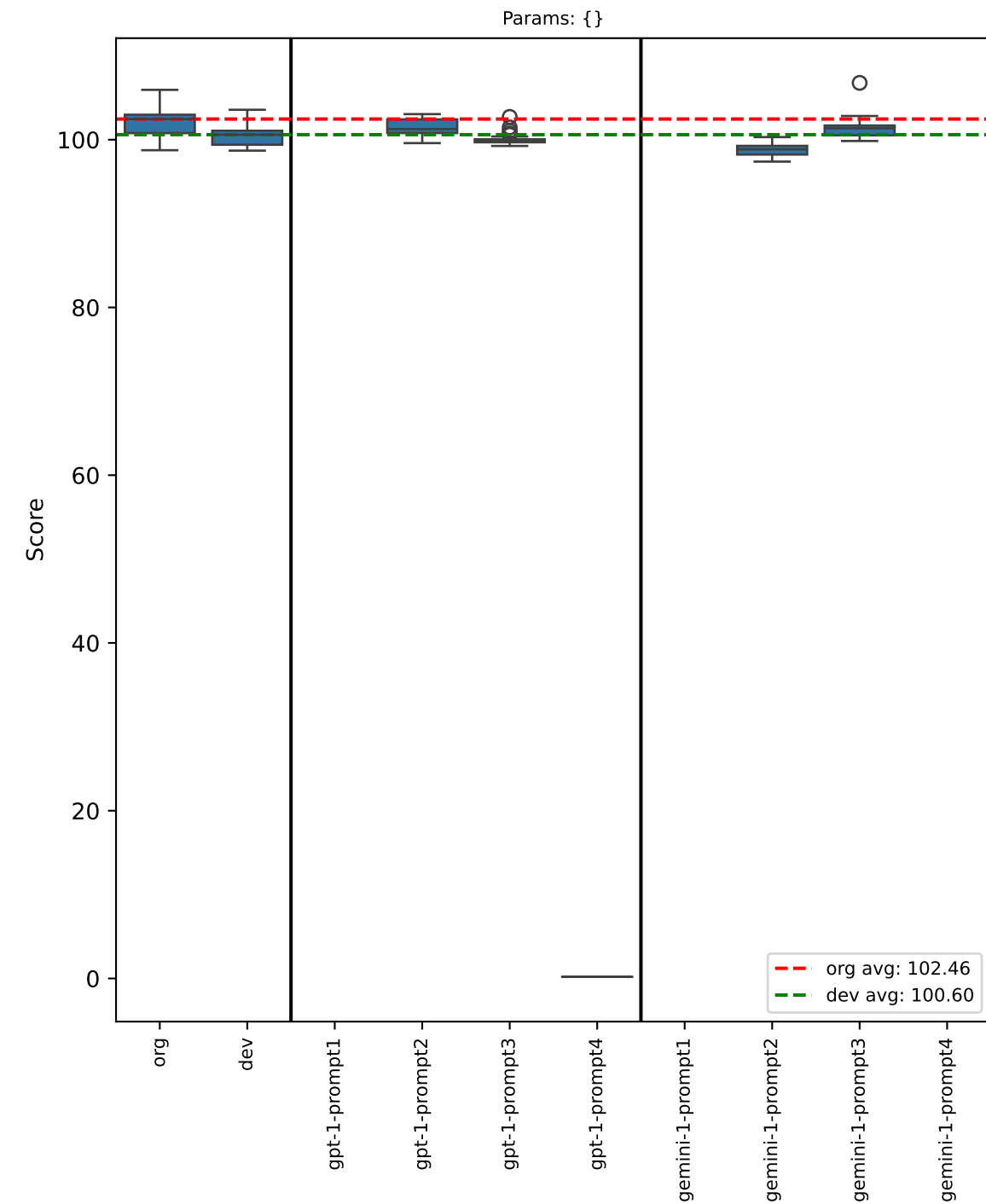


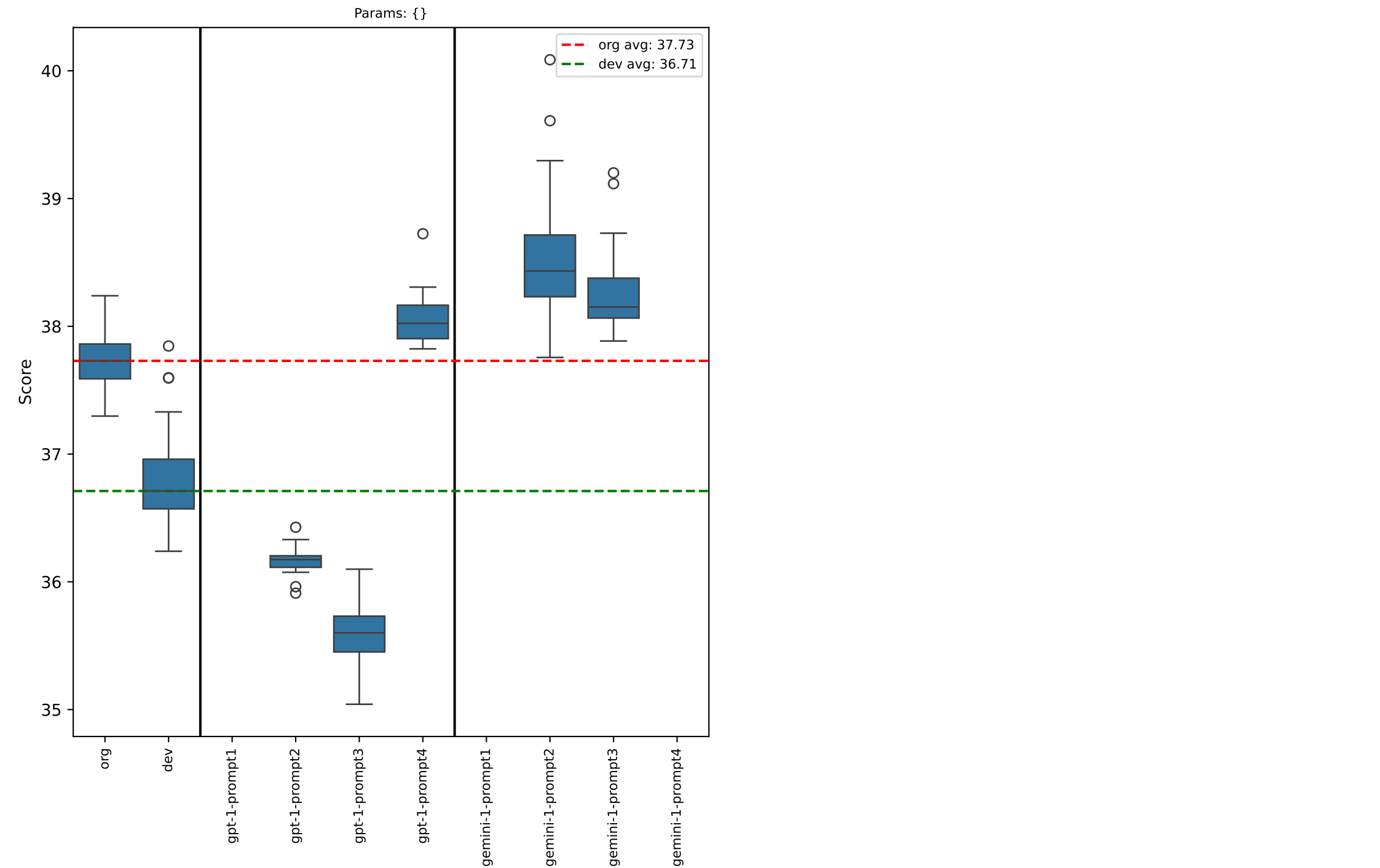




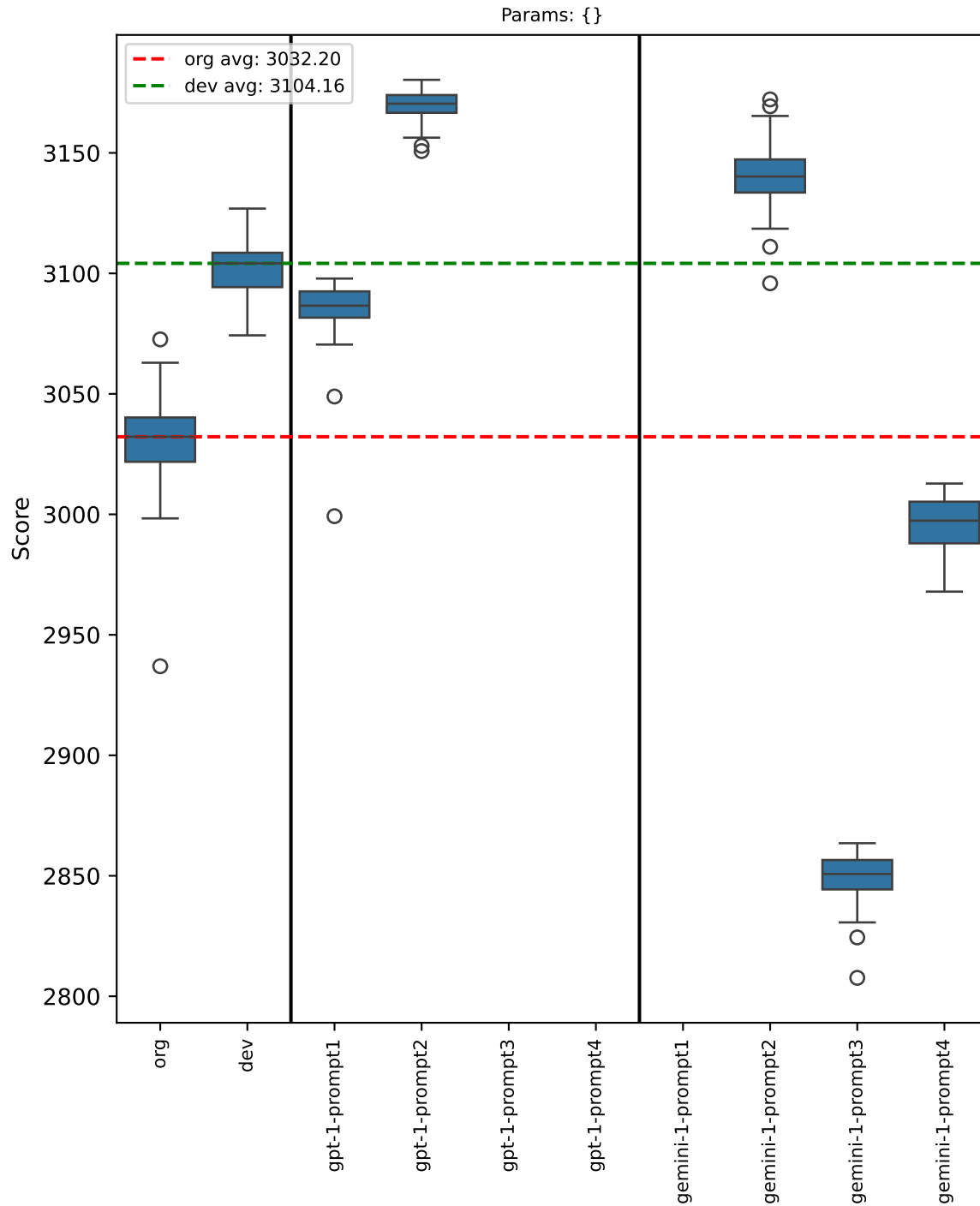


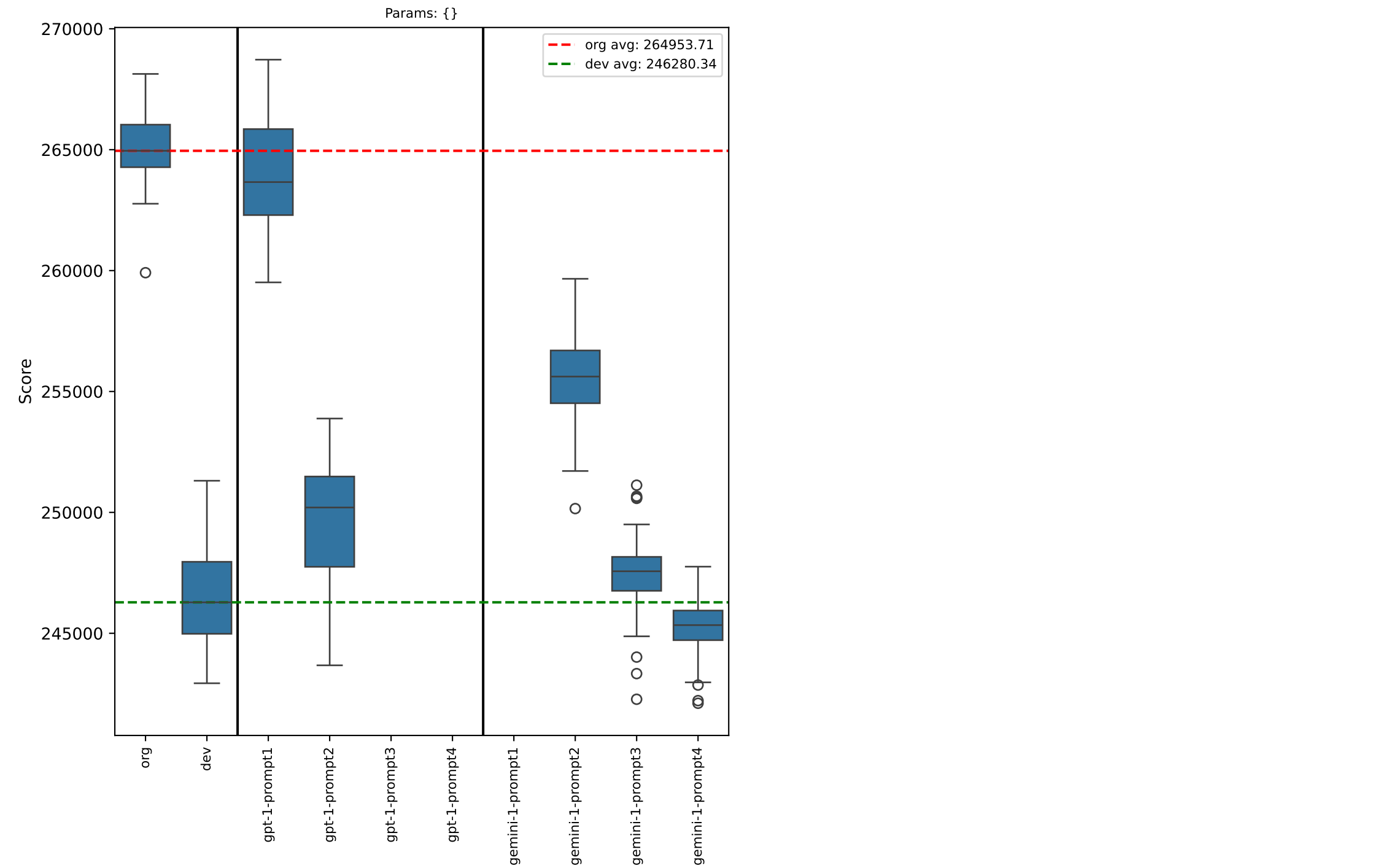




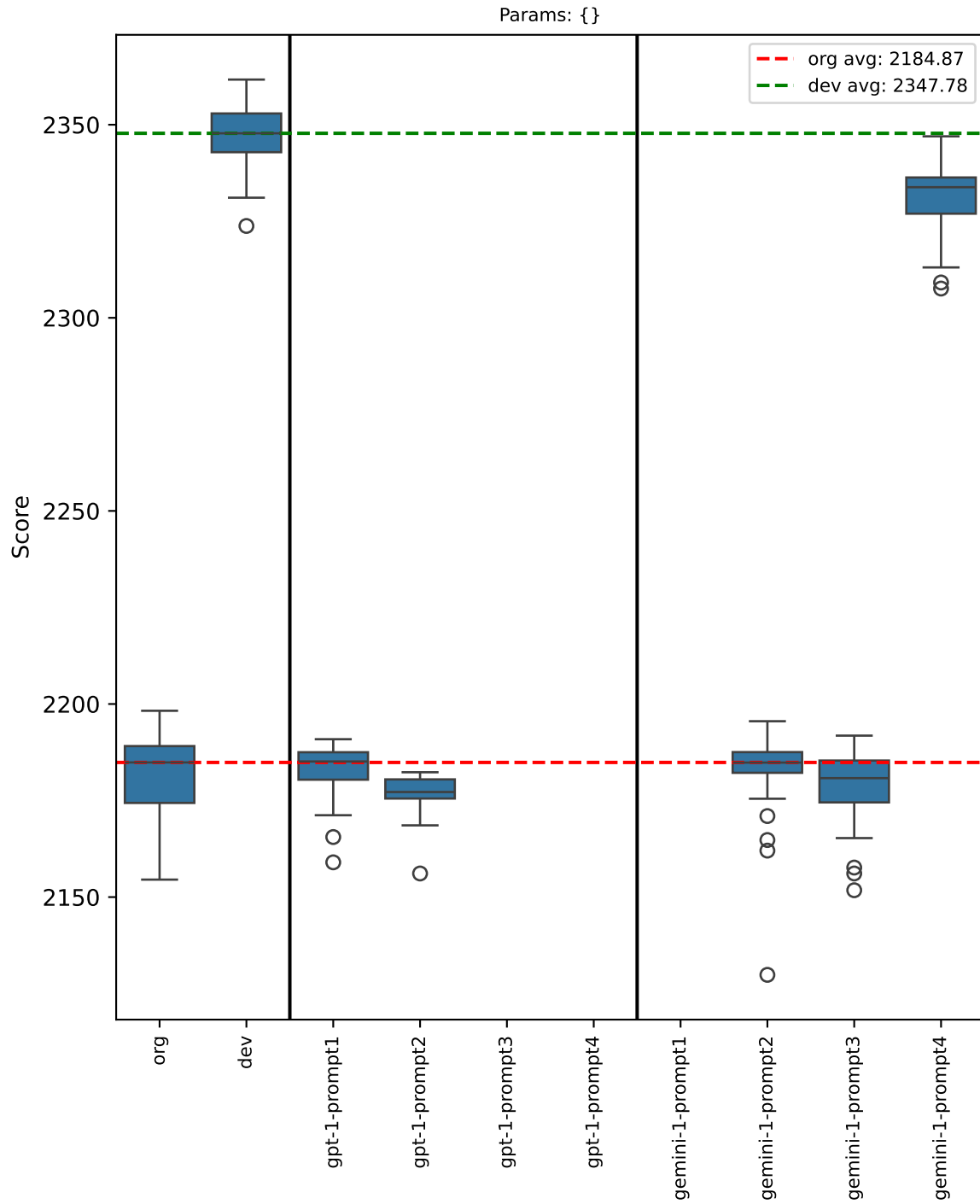


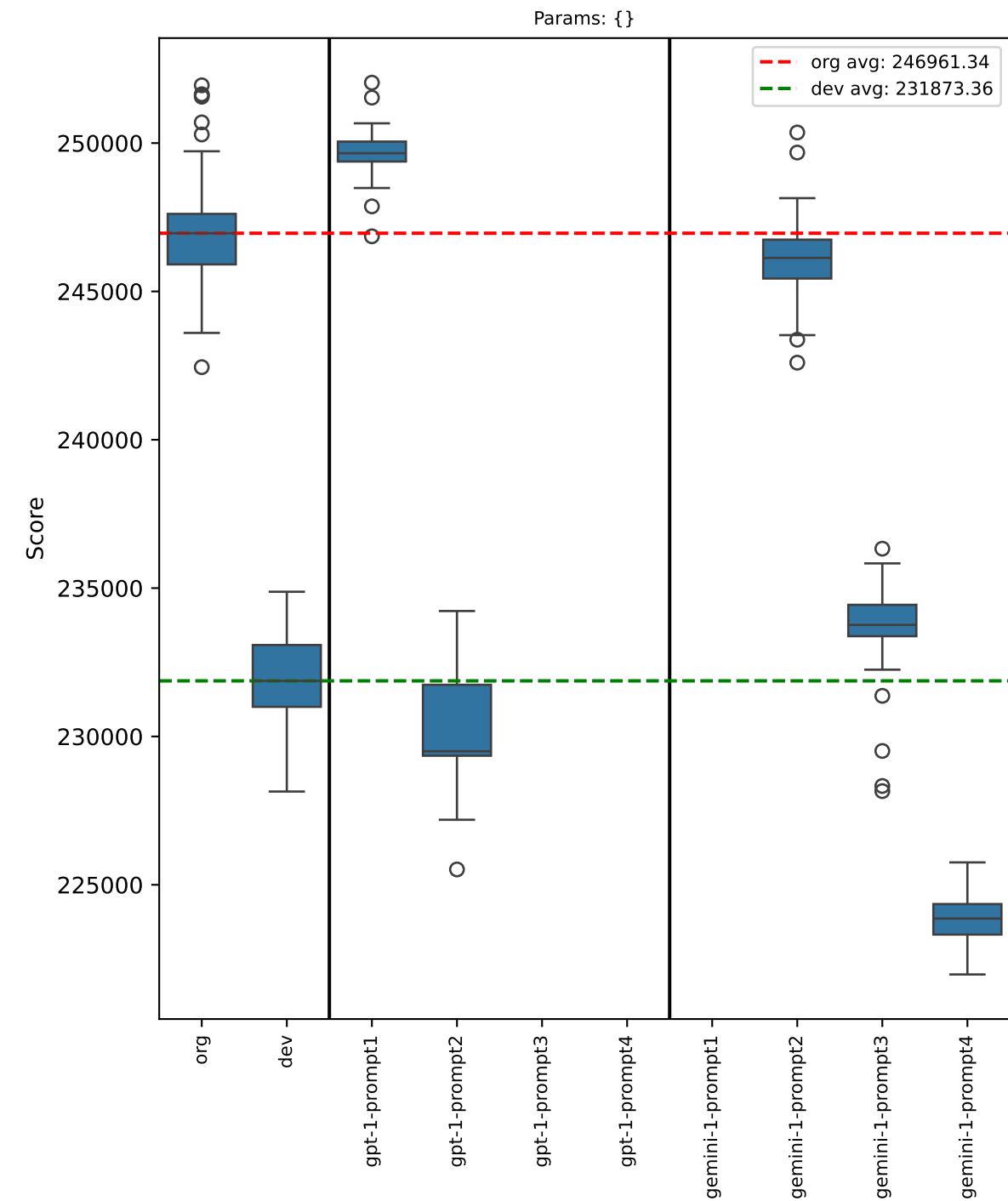
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTIntersects.stIntersectsCrossingLine

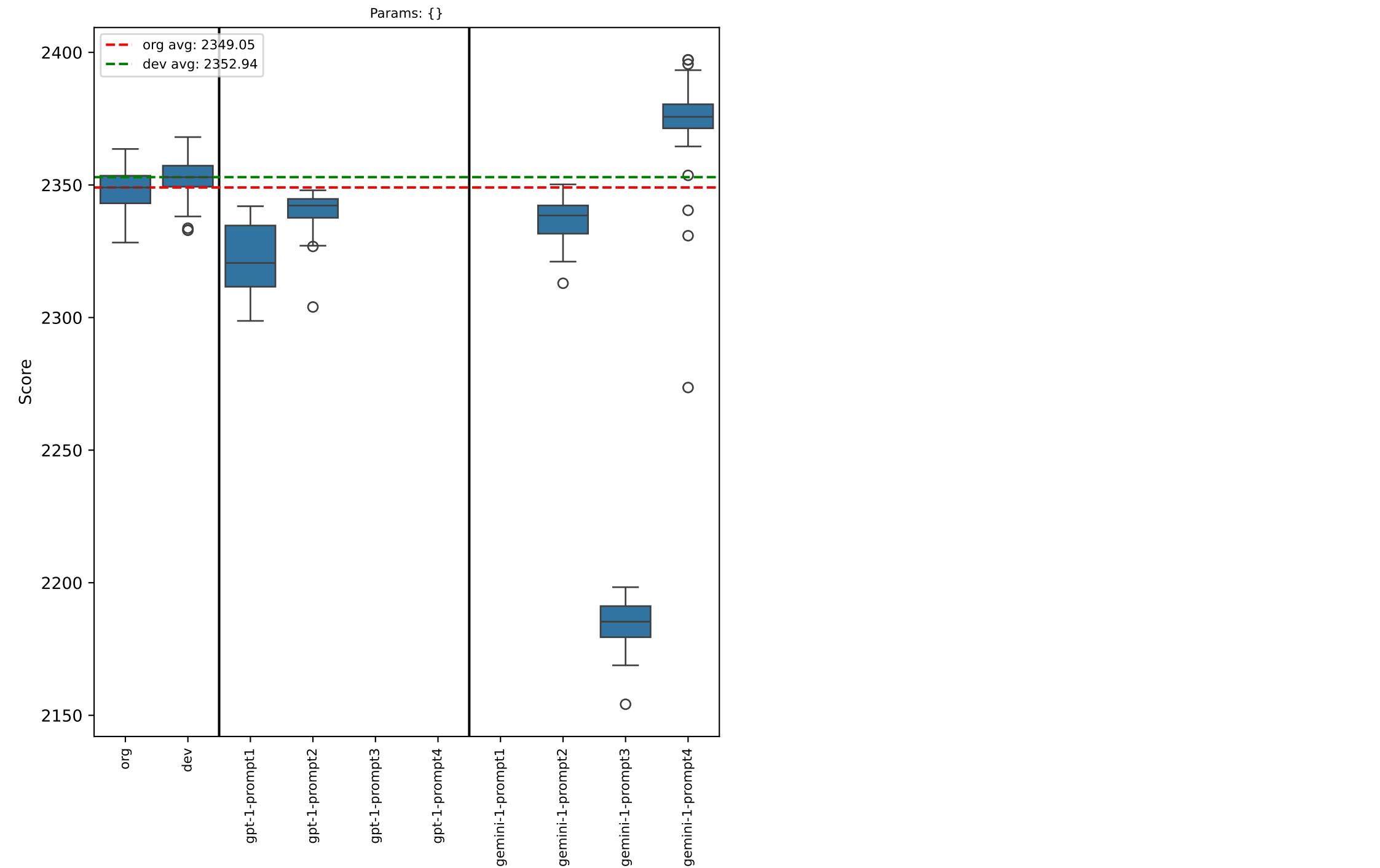


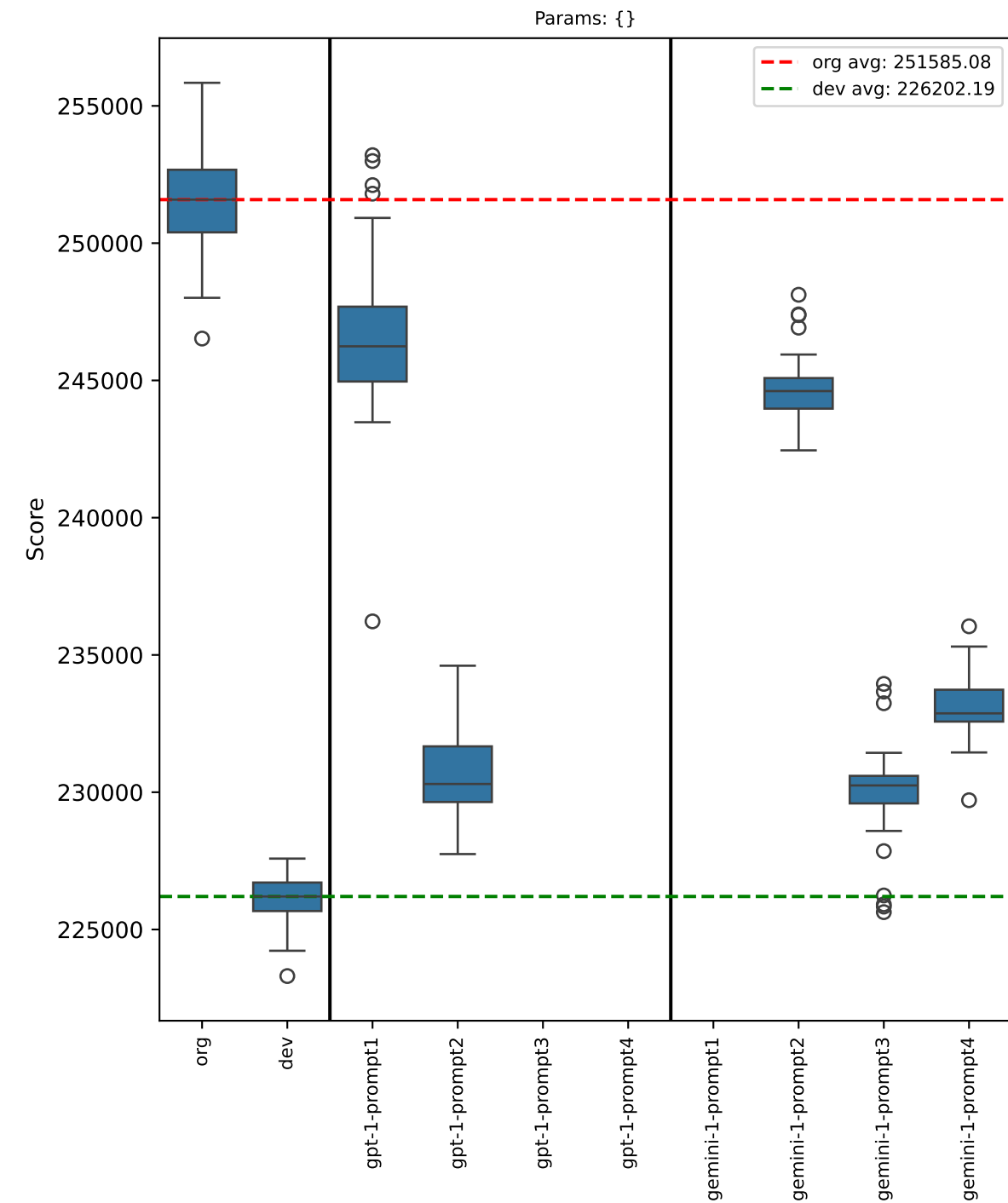


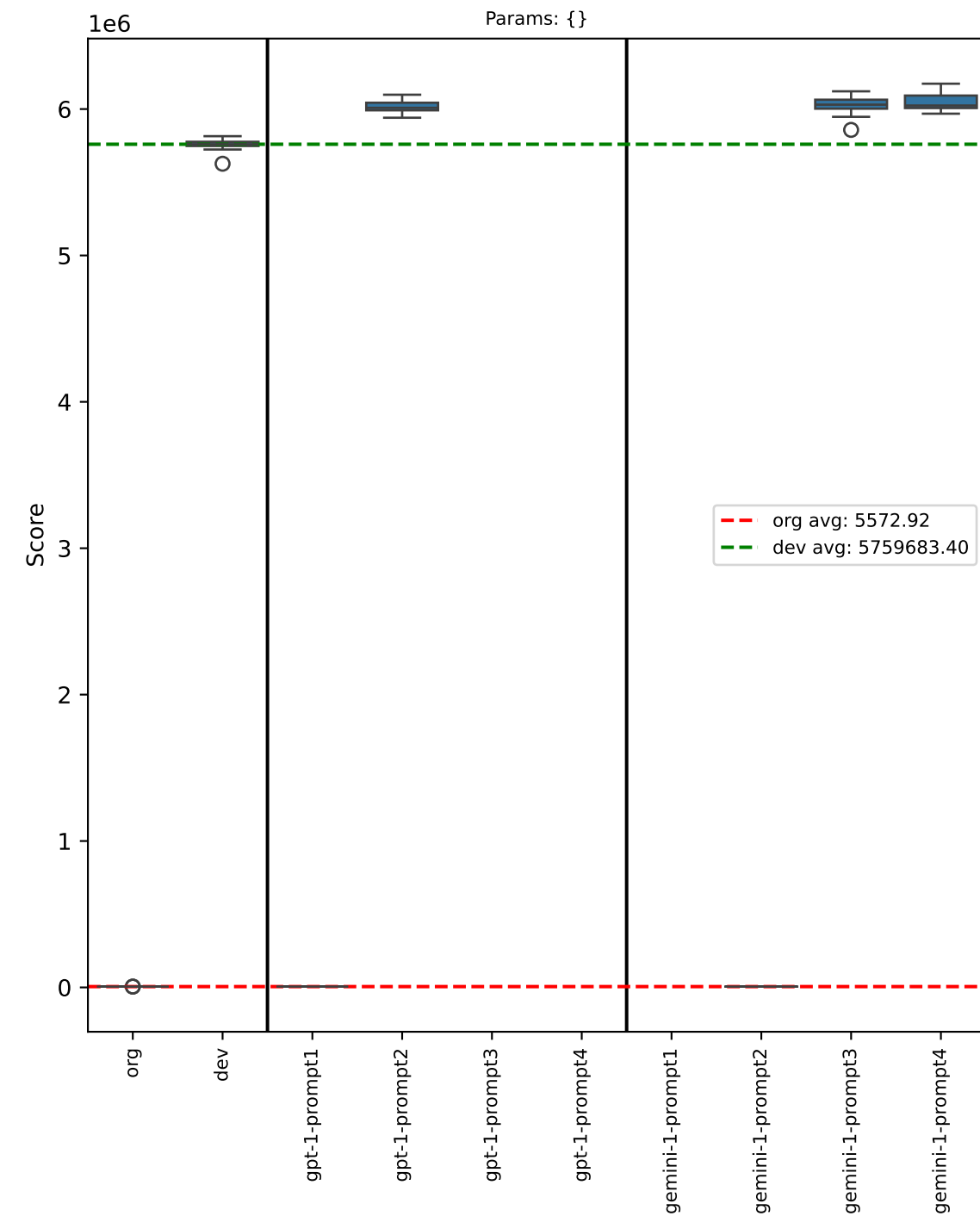
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTIntersects.stIntersectsInnerLine

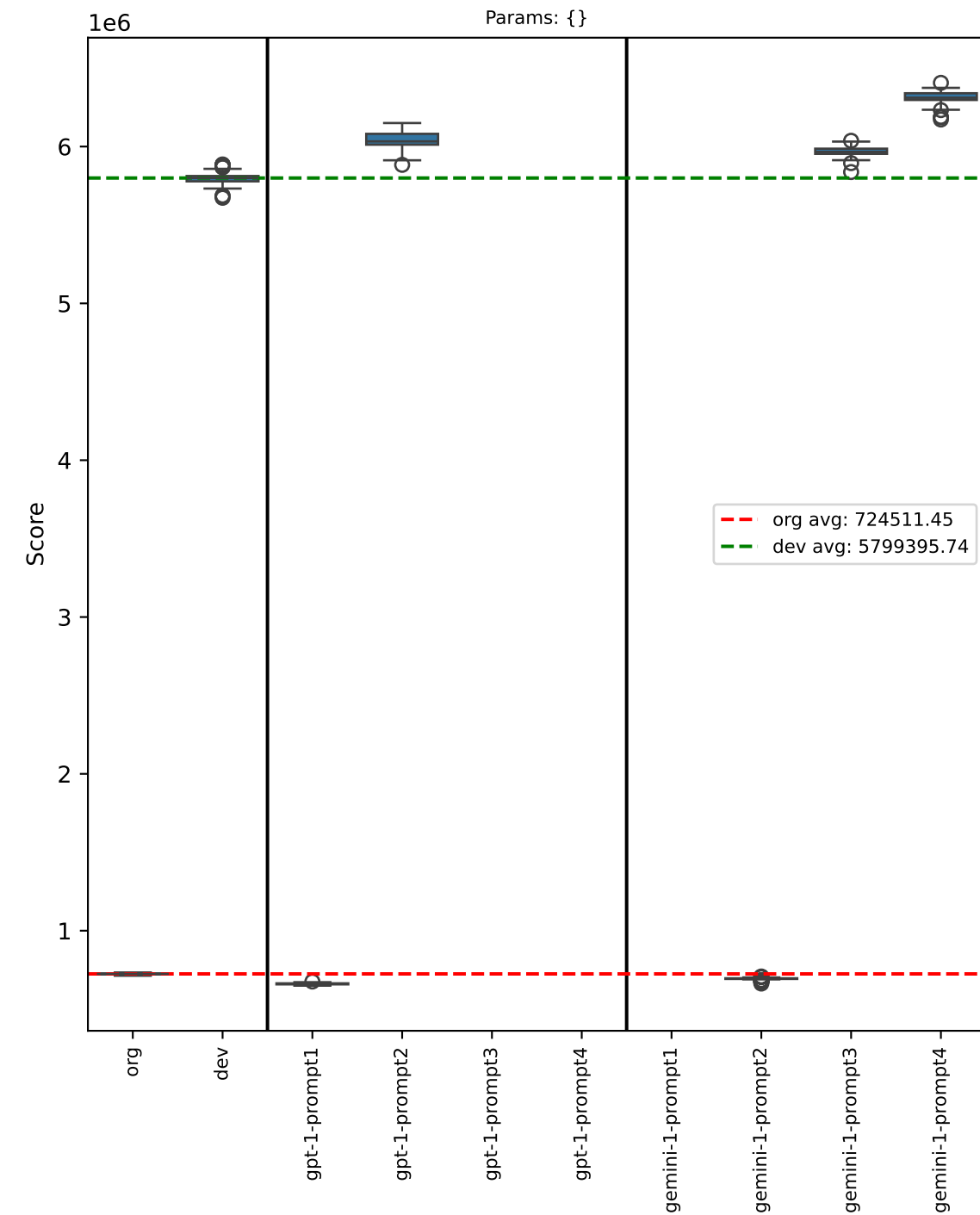




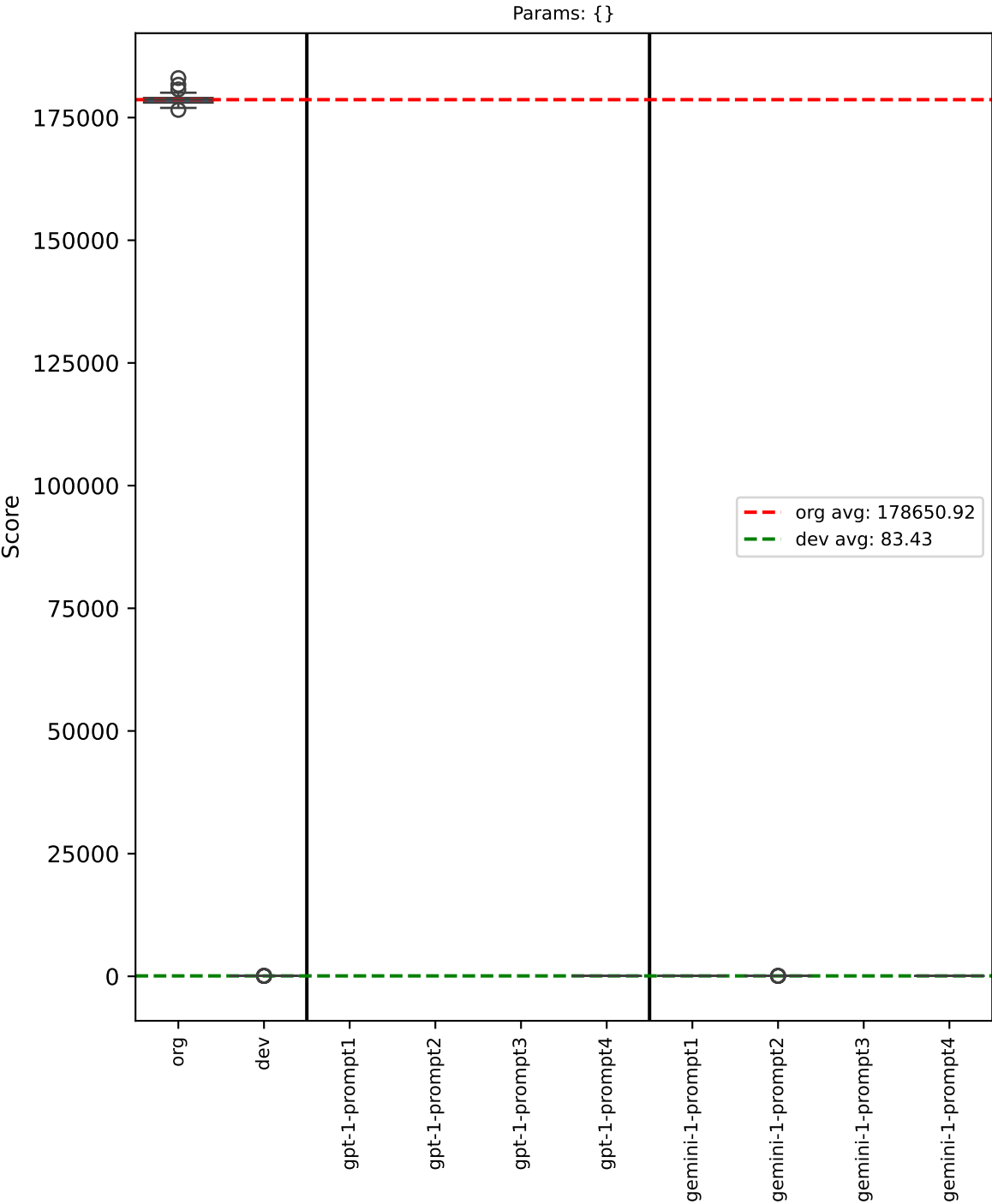




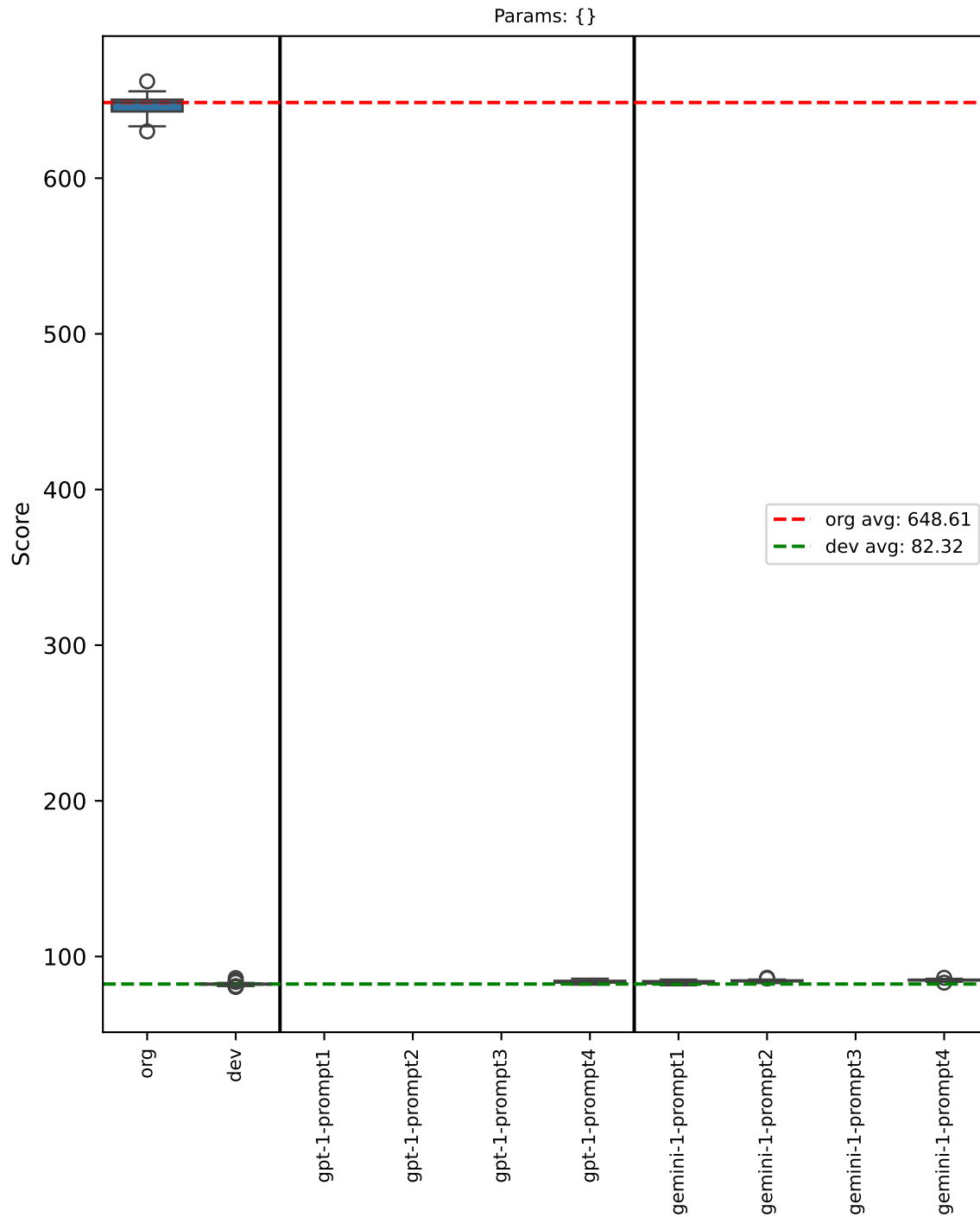




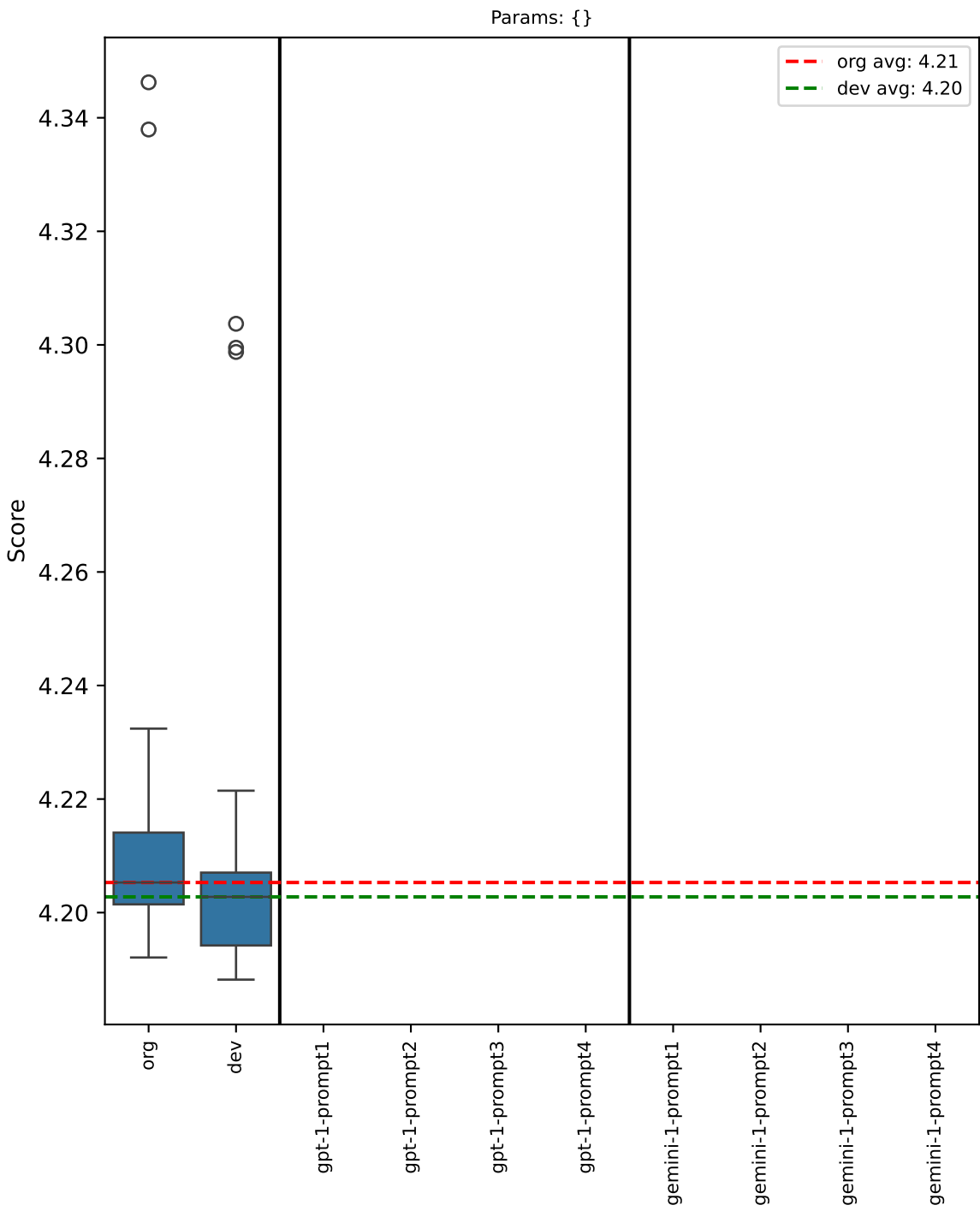
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTXMin.complexGeometry



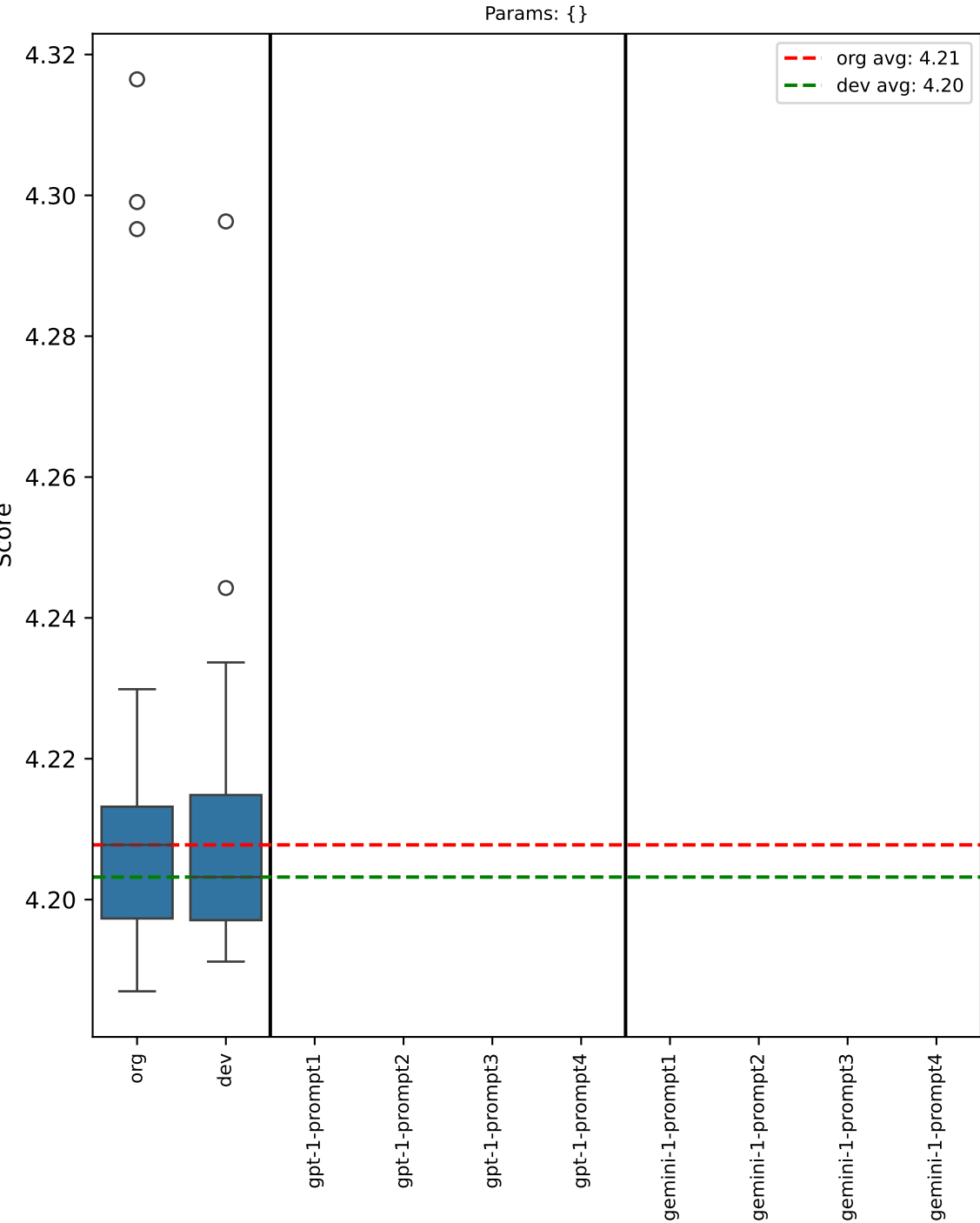
Benchmark: com.facebook.presto.plugin.geospatial.BenchmarkSTXMin.simpleGeometry



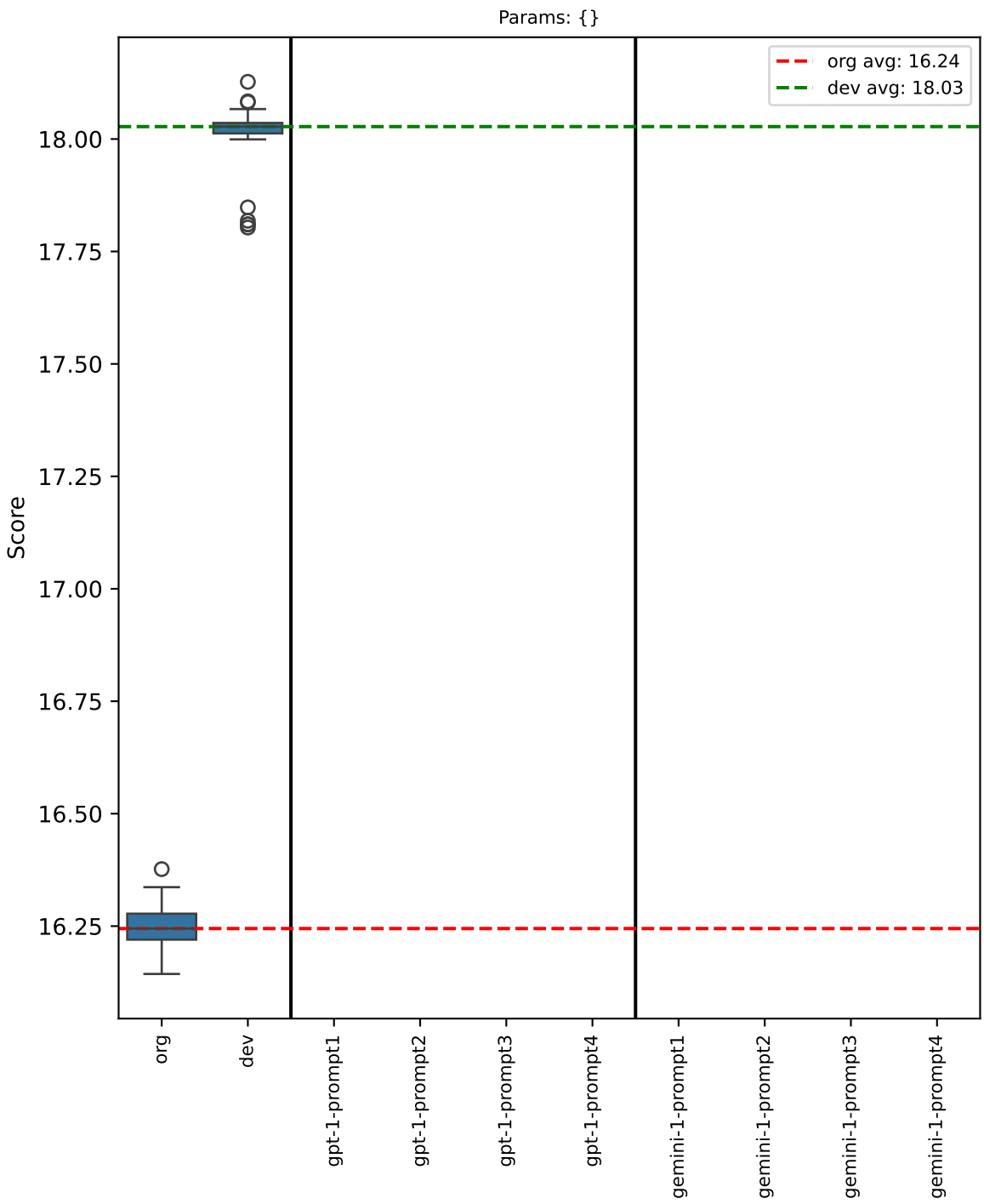
Benchmark: com.facebook.presto.spi.block.BenchmarkComputePosition.computePositionWithBitShifting



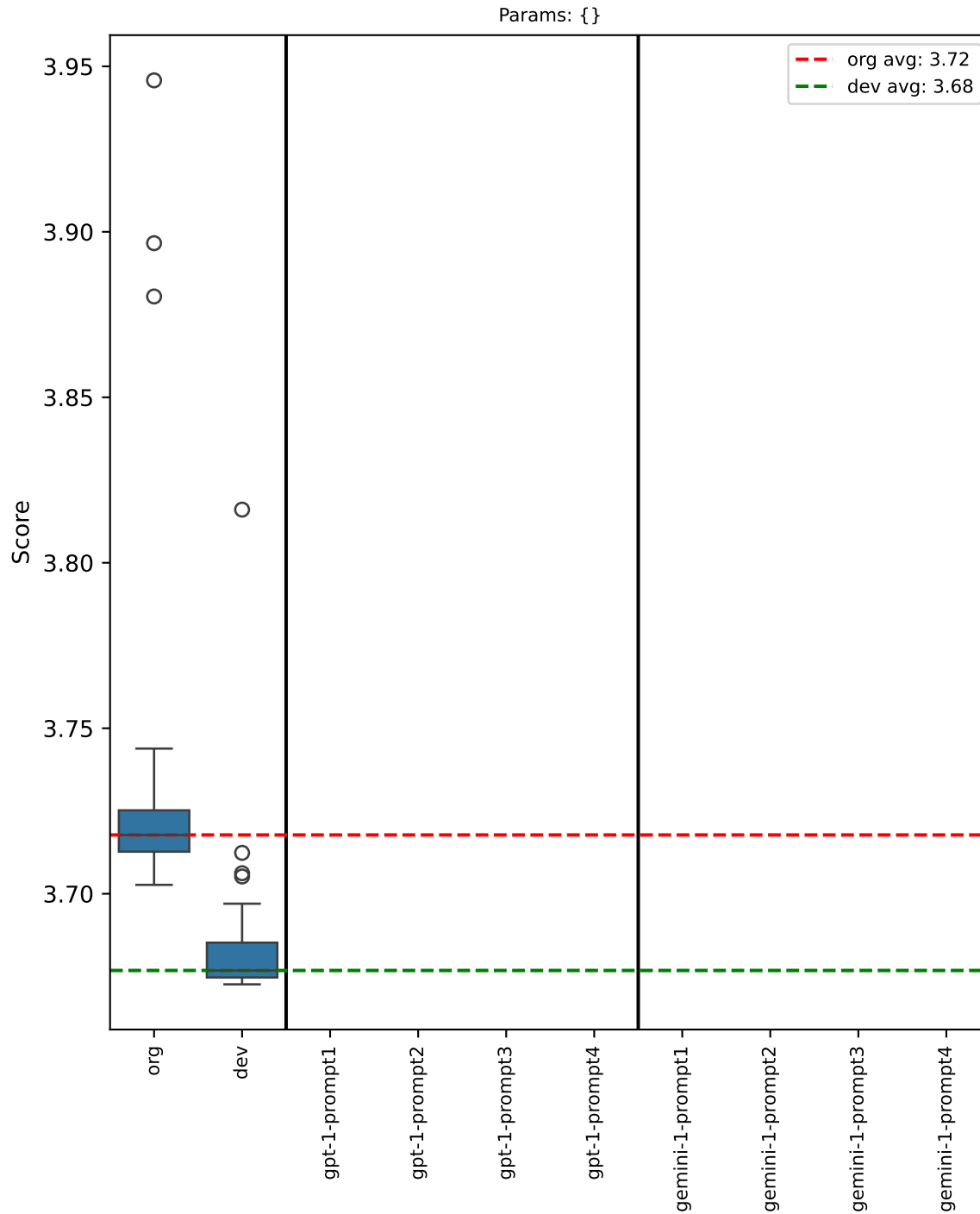
Benchmark: com.facebook.presto.spi.block.BenchmarkComputePosition.computePositionWithDivision



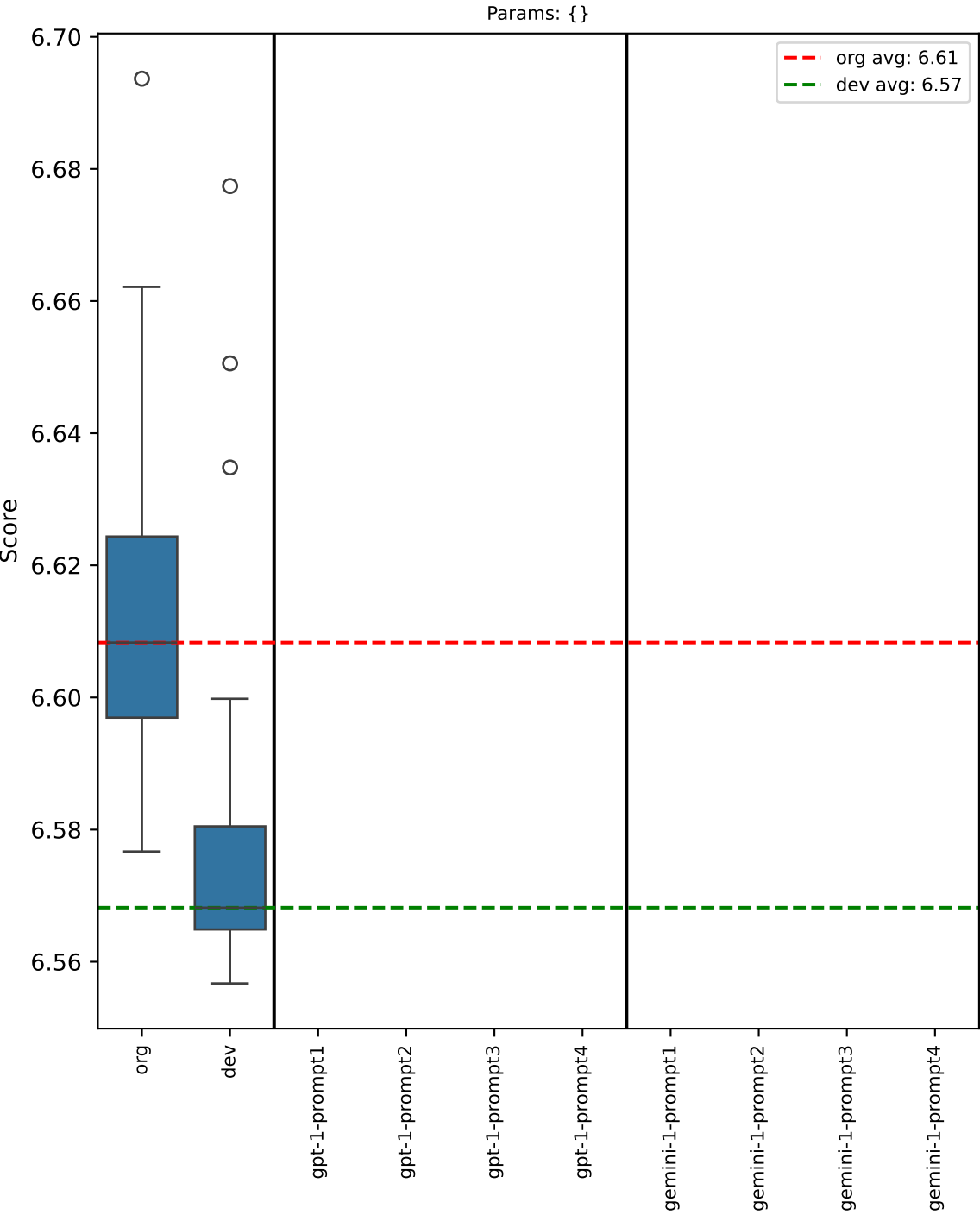
Benchmark: com.facebook.presto.spi.block.BenchmarkComputePosition.computePositionWithFloorMod



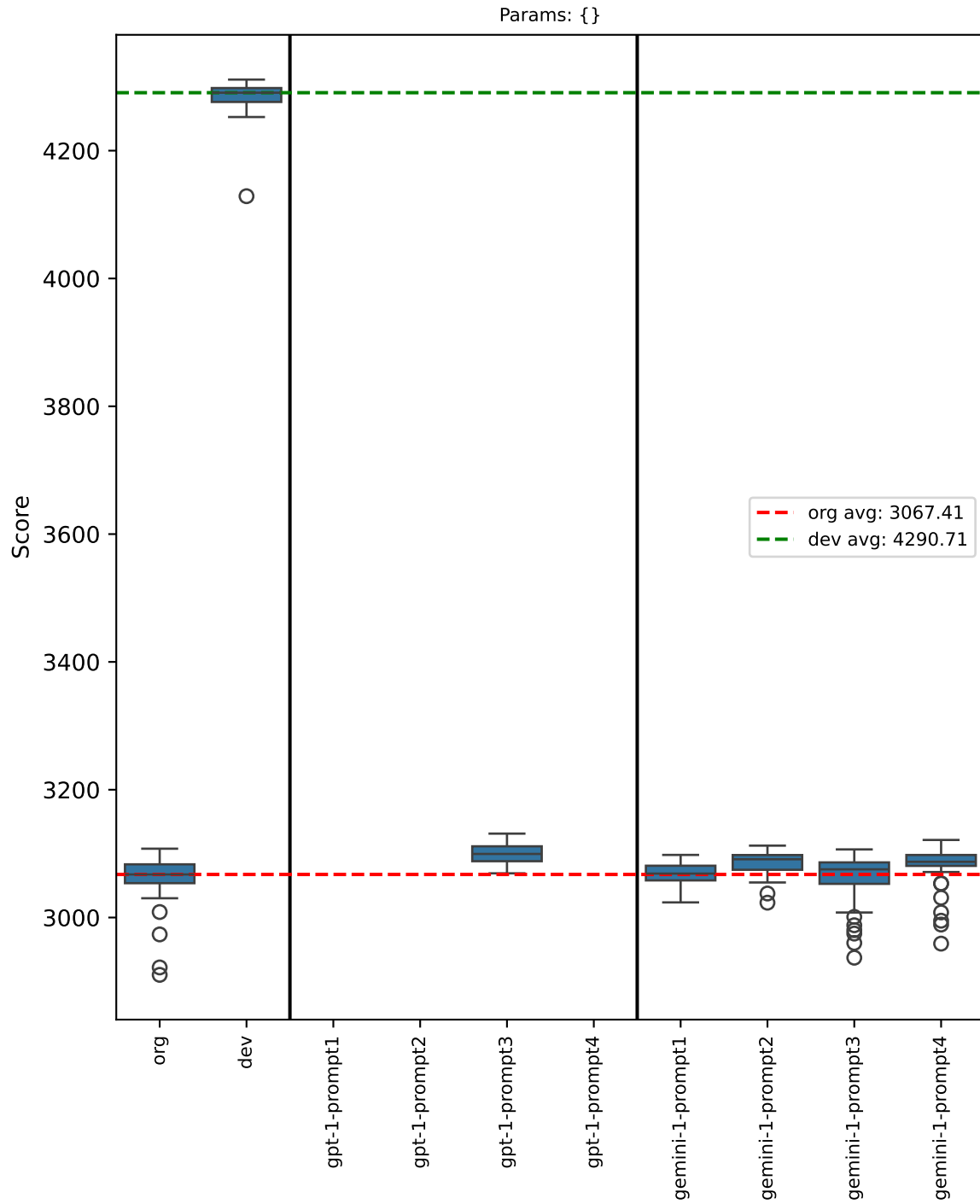
Benchmark: com.facebook.presto.spi.block.BenchmarkComputePosition.computePositionWithMask



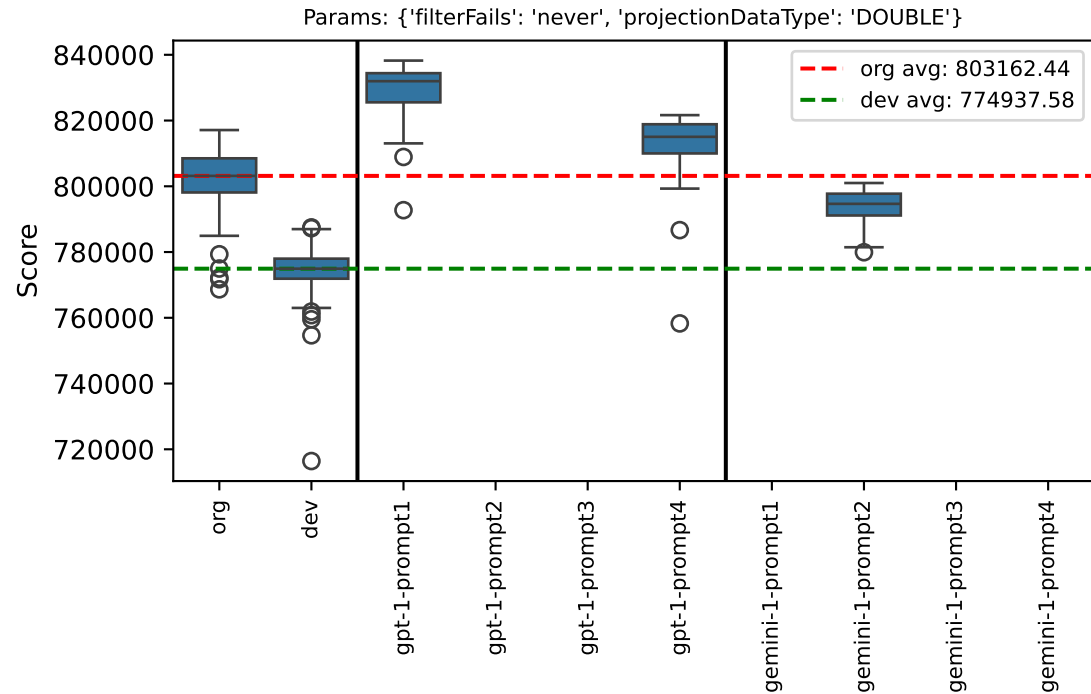
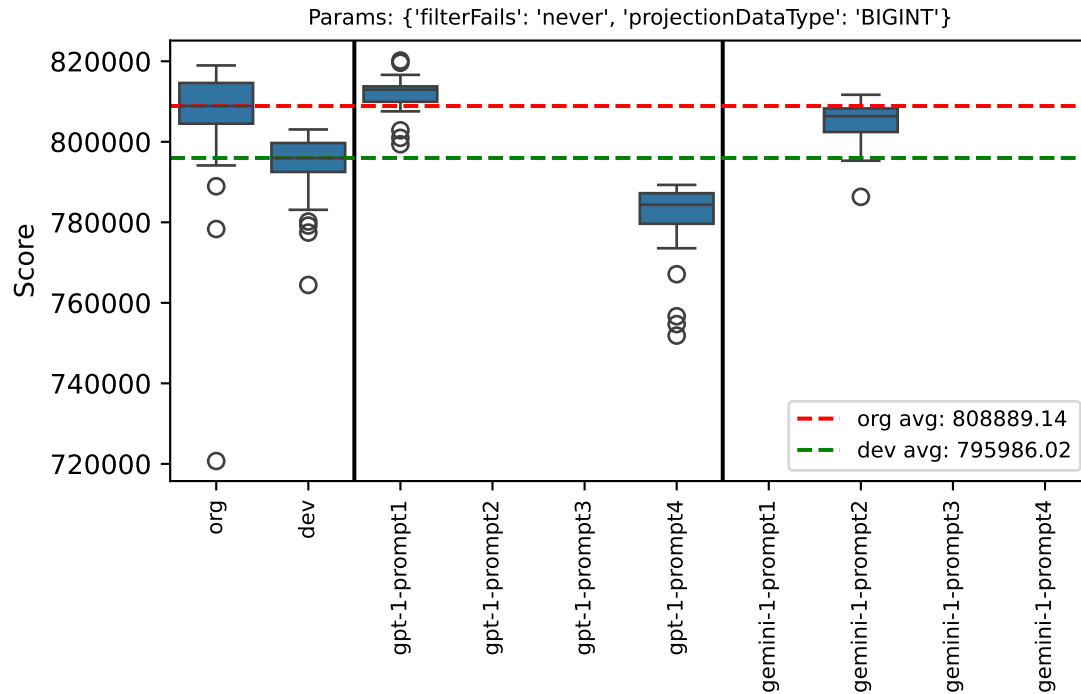
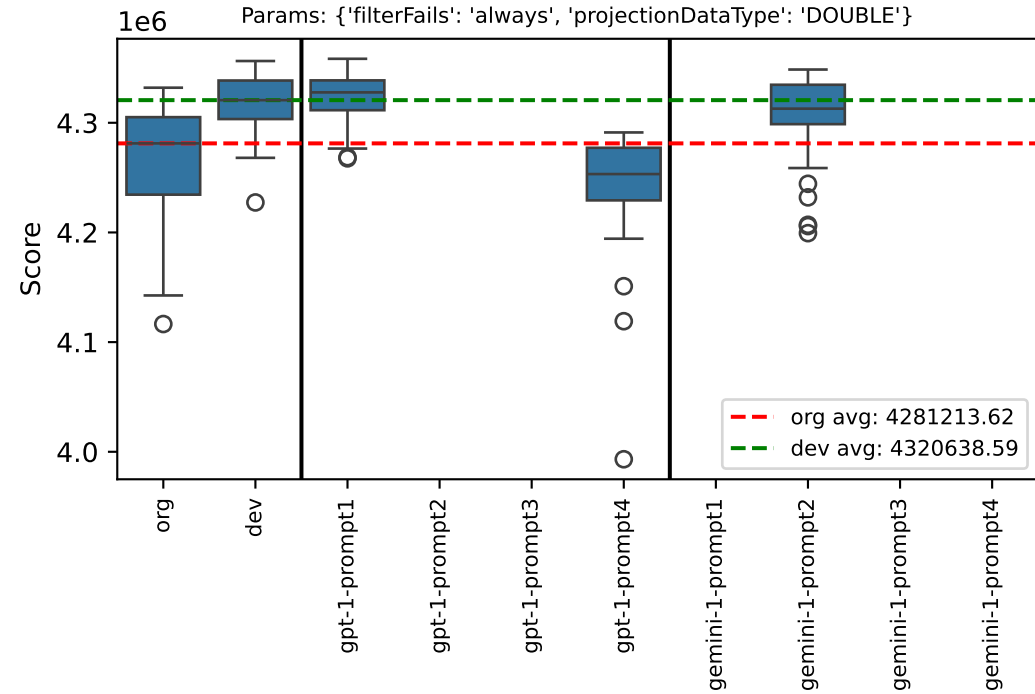
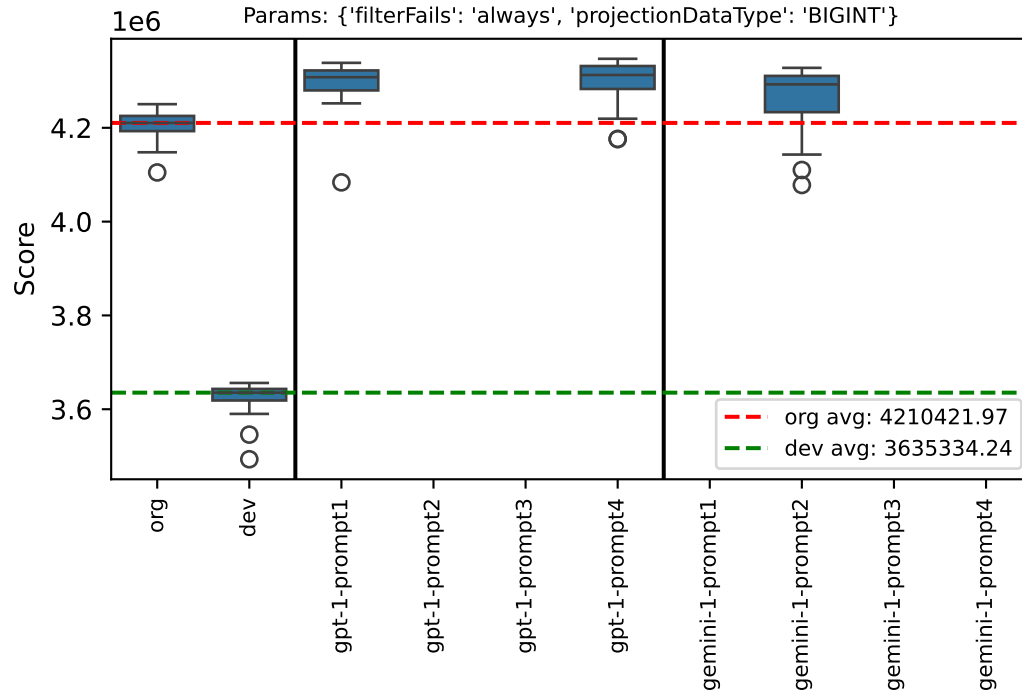
Benchmark: com.facebook.presto.spi.block.BenchmarkComputePosition.computePositionWithMod



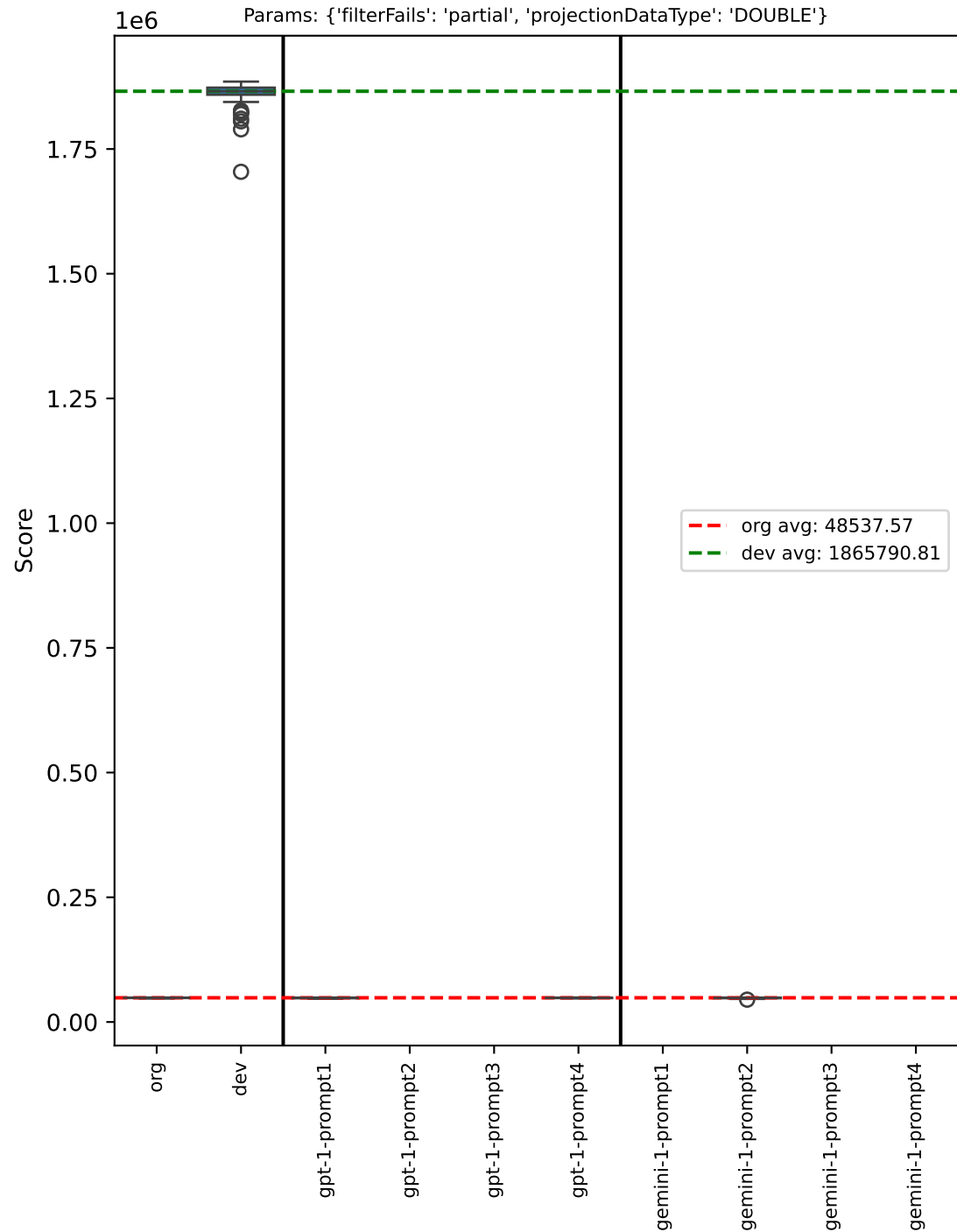
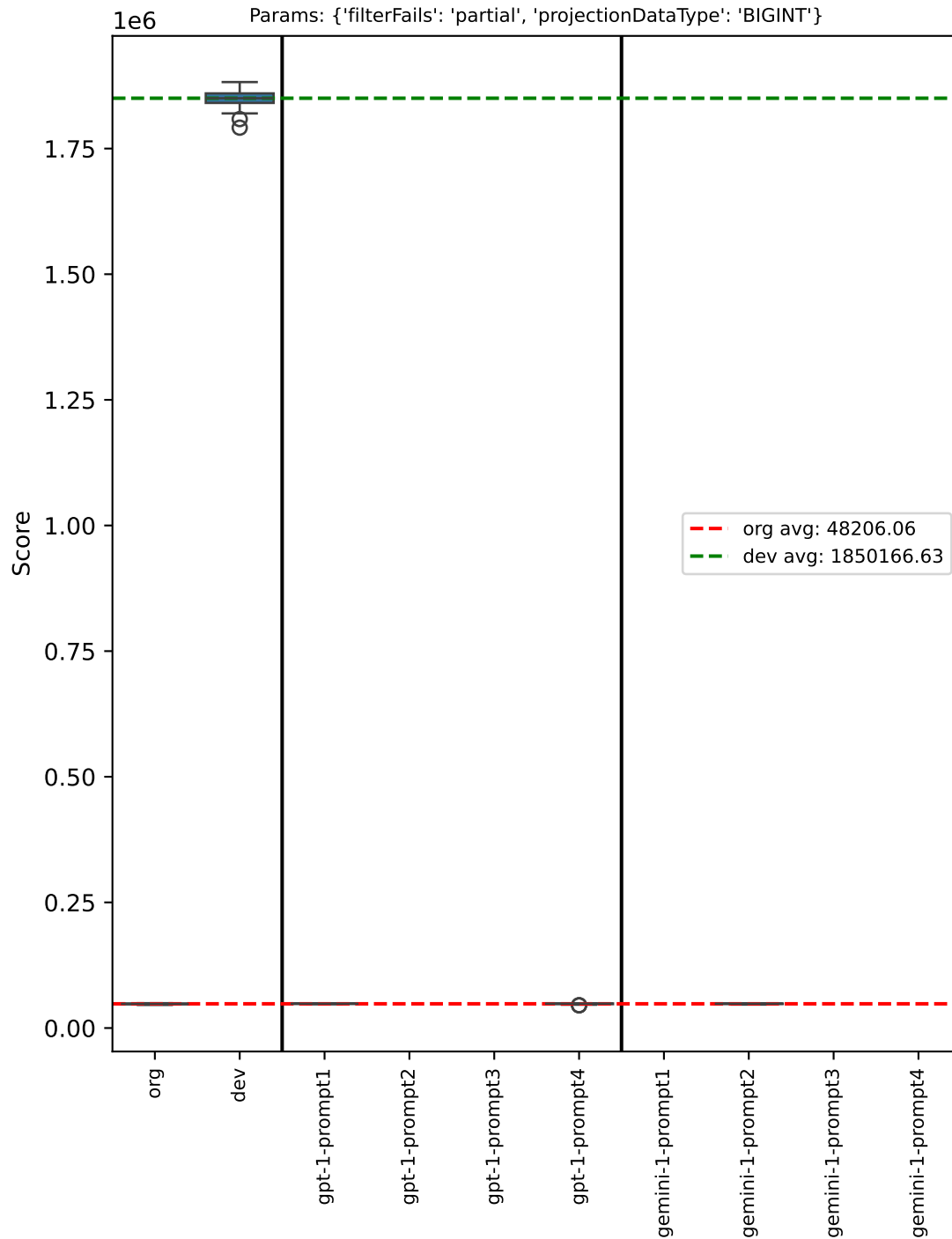
Benchmark: com.facebook.presto.sql.gen.BenchmarkPageProcessor.compiled



Benchmark: com.facebook.presto.sql.gen.BenchmarkPageProcessor.identityProjection

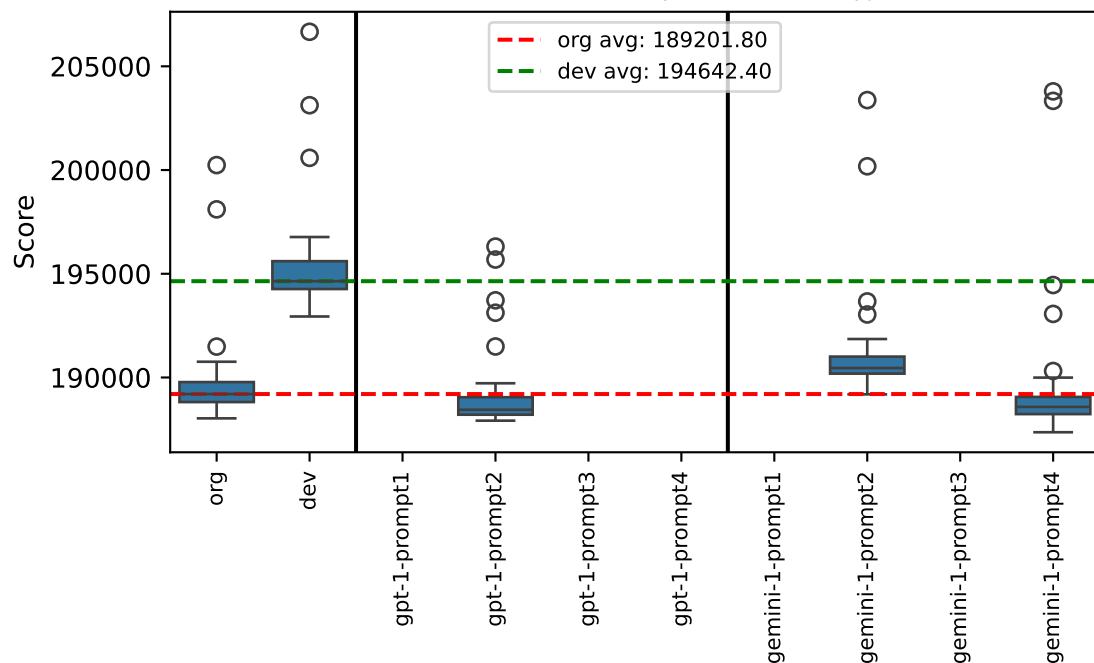


Benchmark: com.facebook.presto.sql.gen.BenchmarkPageProcessor.identityProjection

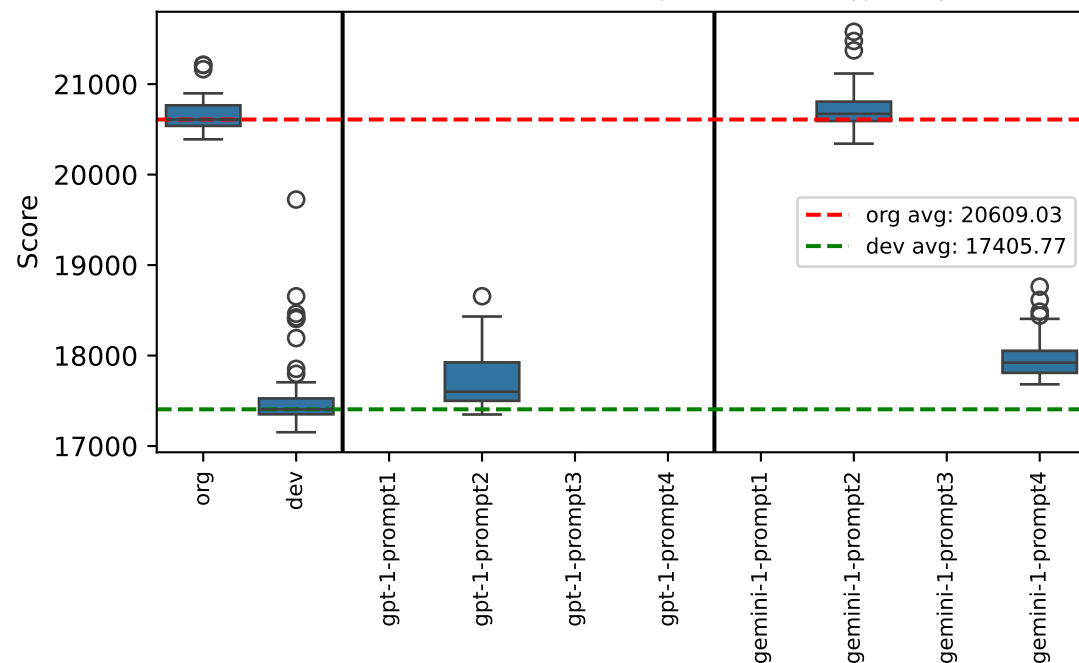


Benchmark: com.facebook.presto.sql.gen.PageProcessorBenchmark.columnOriented

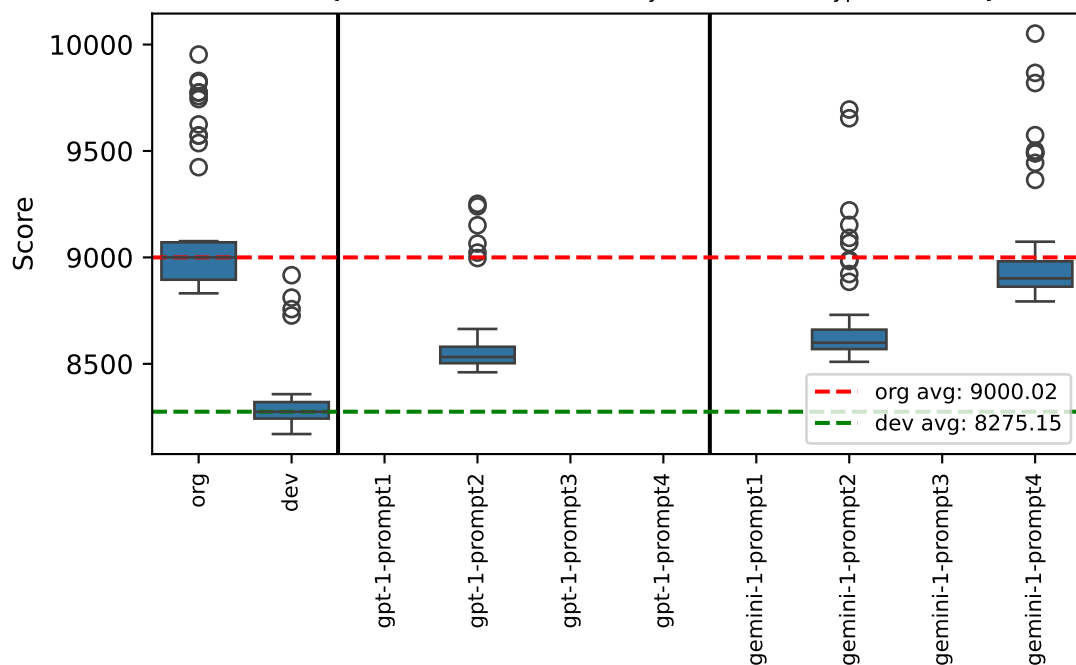
Params: {'columnCount': '2', 'dictionaryBlocks': 'false', 'type': 'varchar'}



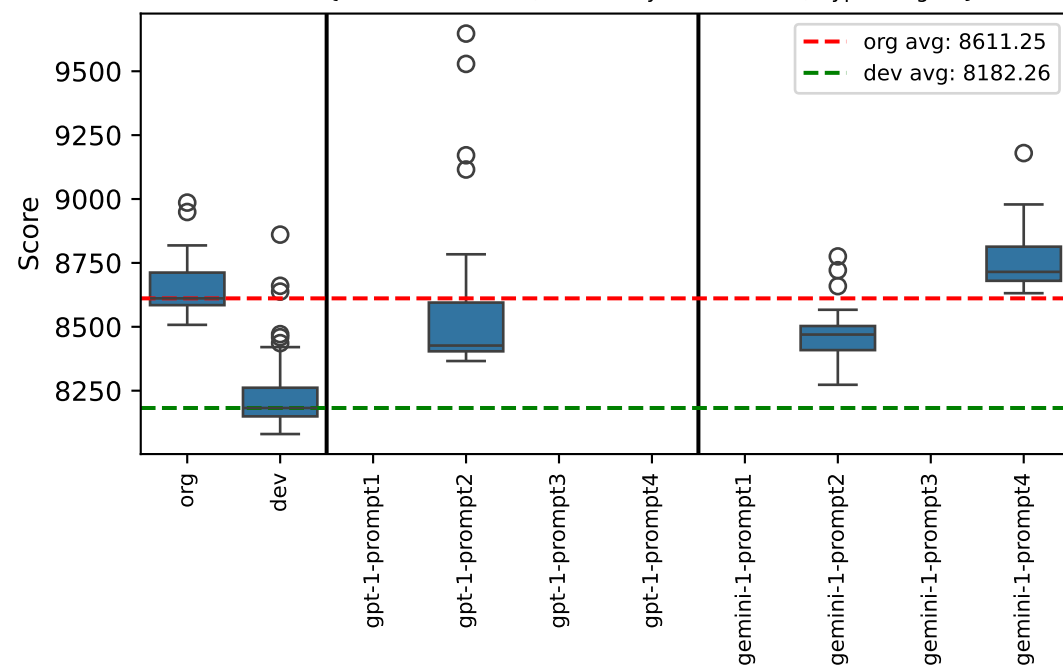
Params: {'columnCount': '2', 'dictionaryBlocks': 'false', 'type': 'bigint'}



Params: {'columnCount': '2', 'dictionaryBlocks': 'true', 'type': 'varchar'}

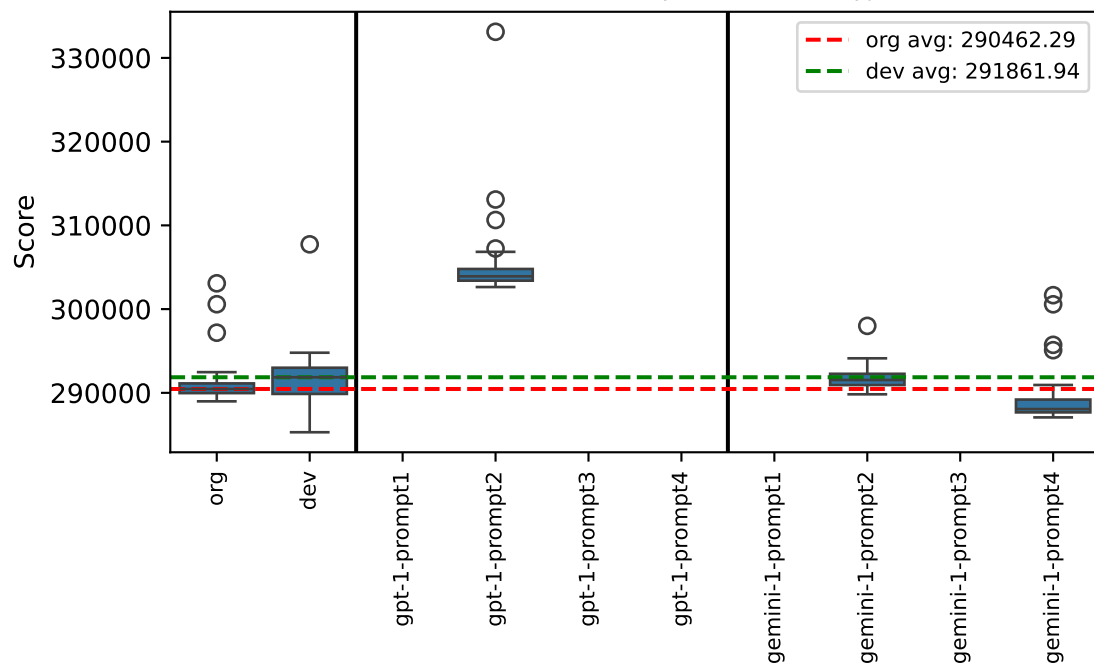


Params: {'columnCount': '2', 'dictionaryBlocks': 'true', 'type': 'bigint'}

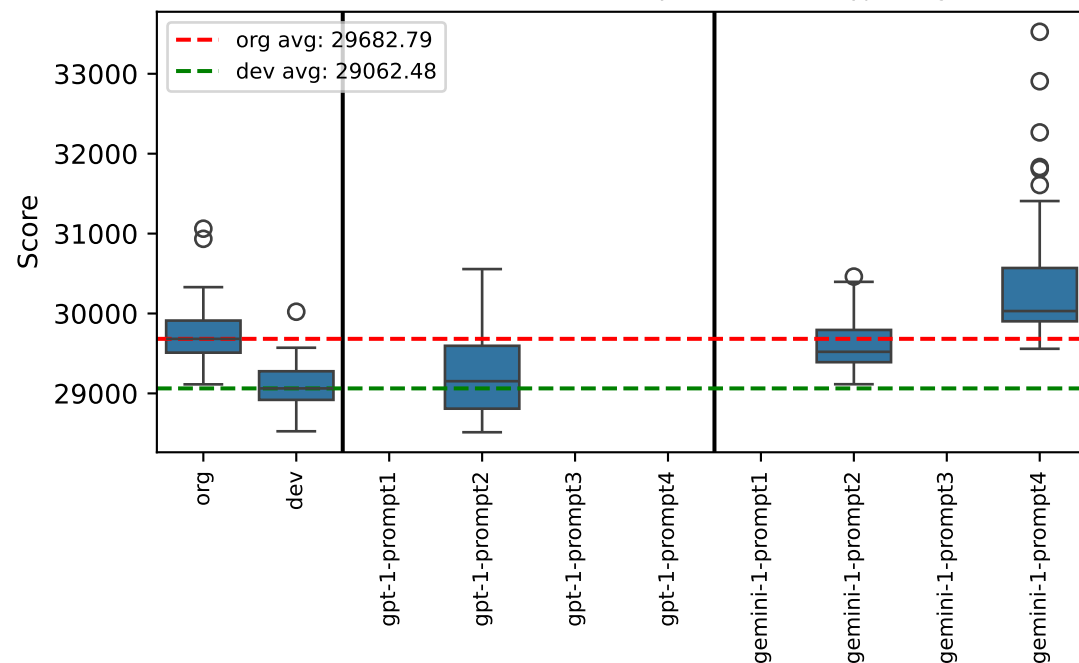


Benchmark: com.facebook.presto.sql.gen.PageProcessorBenchmark.columnOriented

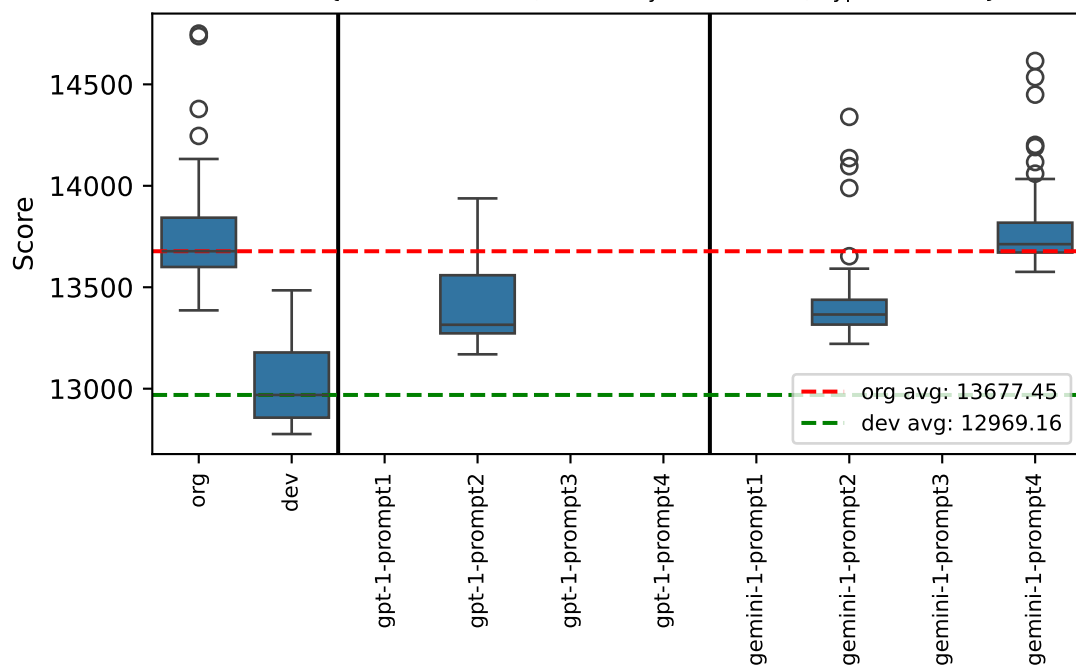
Params: {'columnCount': '4', 'dictionaryBlocks': 'false', 'type': 'varchar'}



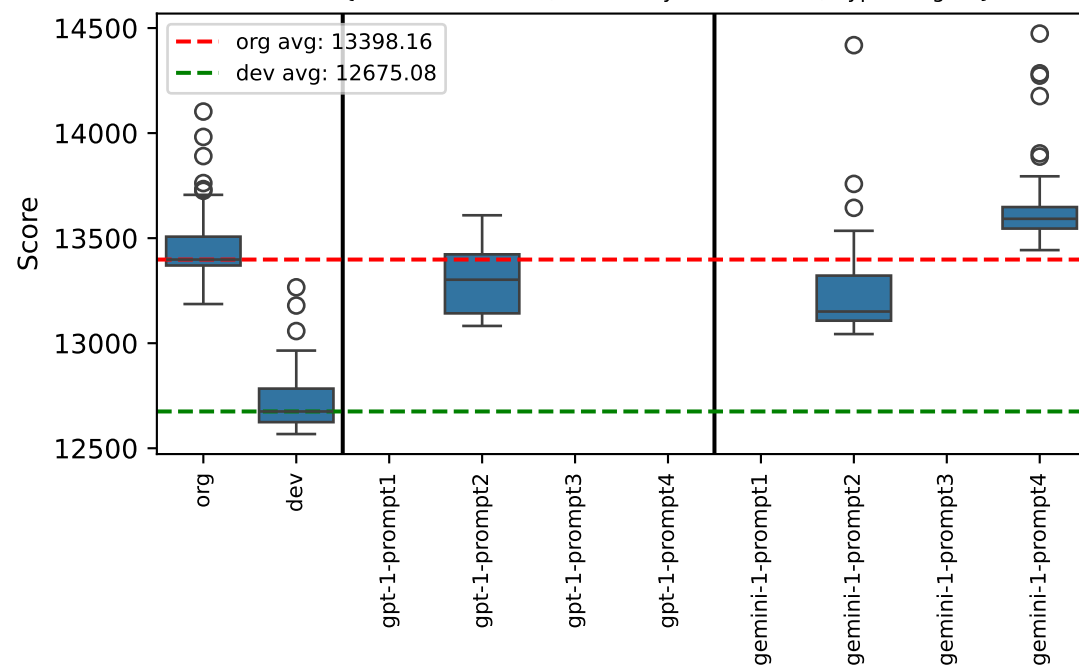
Params: {'columnCount': '4', 'dictionaryBlocks': 'false', 'type': 'bigint'}



Params: {'columnCount': '4', 'dictionaryBlocks': 'true', 'type': 'varchar'}

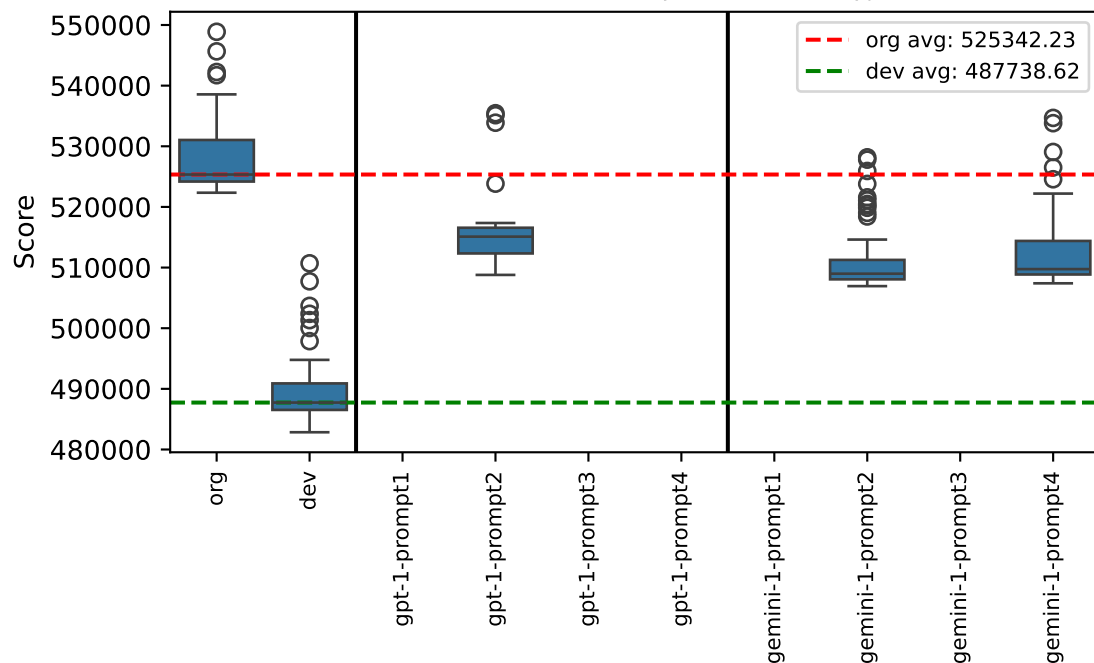


Params: {'columnCount': '4', 'dictionaryBlocks': 'true', 'type': 'bigint'}

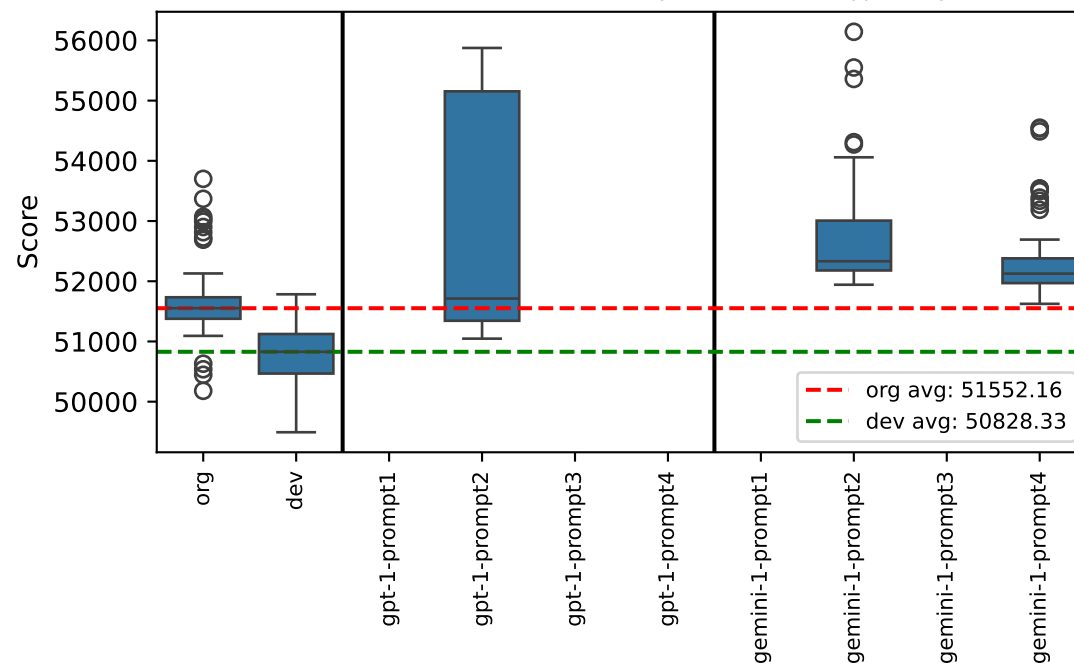


Benchmark: com.facebook.presto.sql.gen.PageProcessorBenchmark.columnOriented

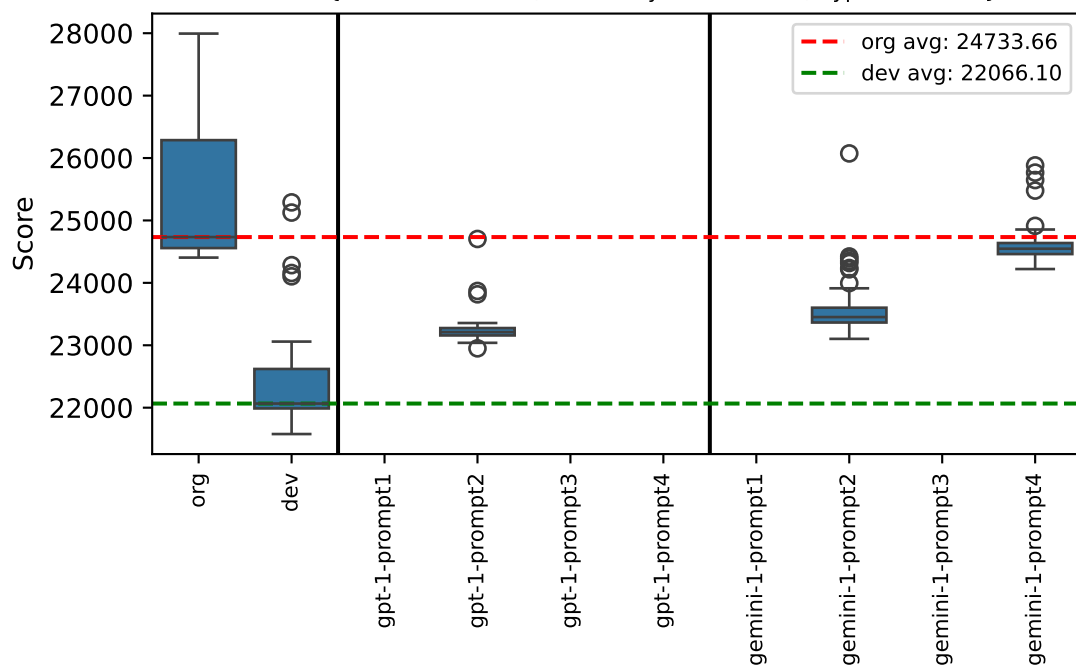
Params: {'columnCount': '8', 'dictionaryBlocks': 'false', 'type': 'varchar'}



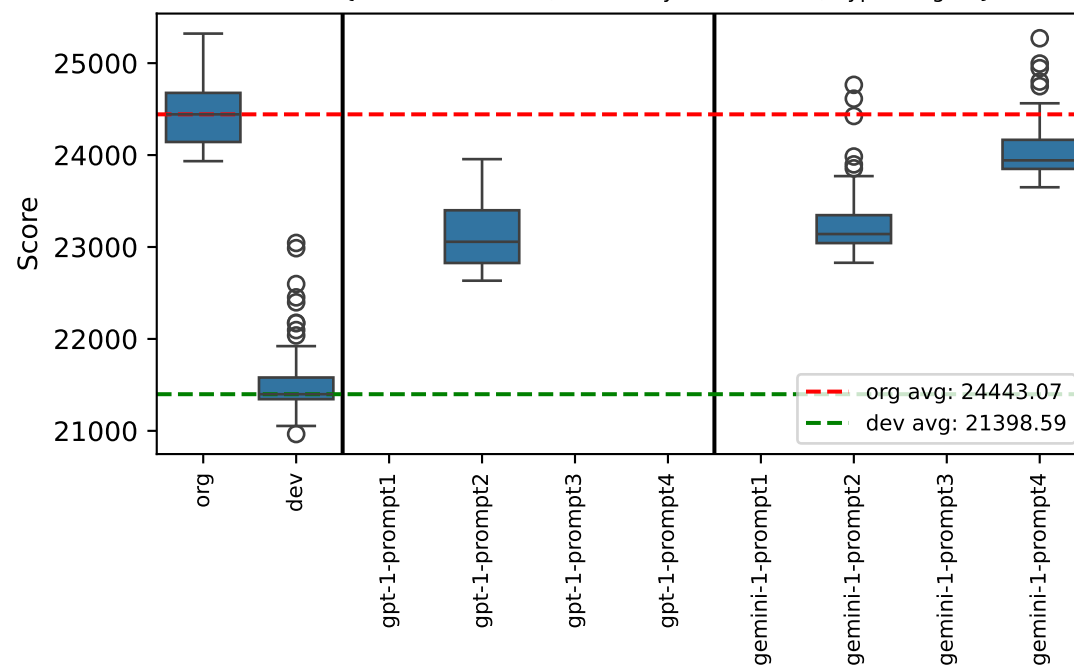
Params: {'columnCount': '8', 'dictionaryBlocks': 'false', 'type': 'bigint'}



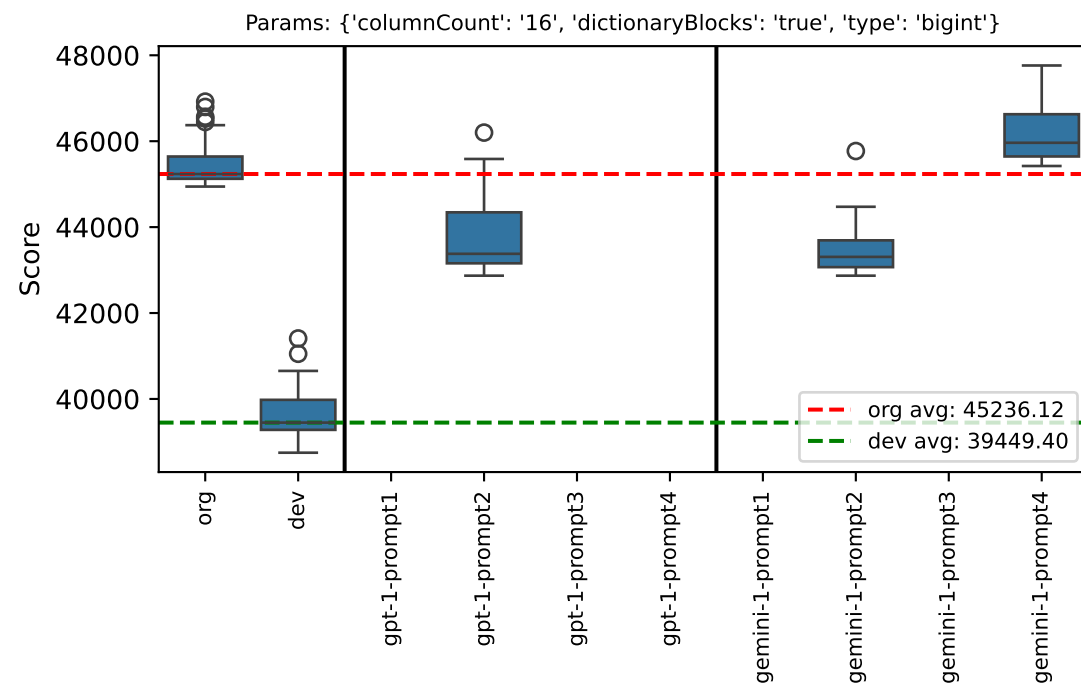
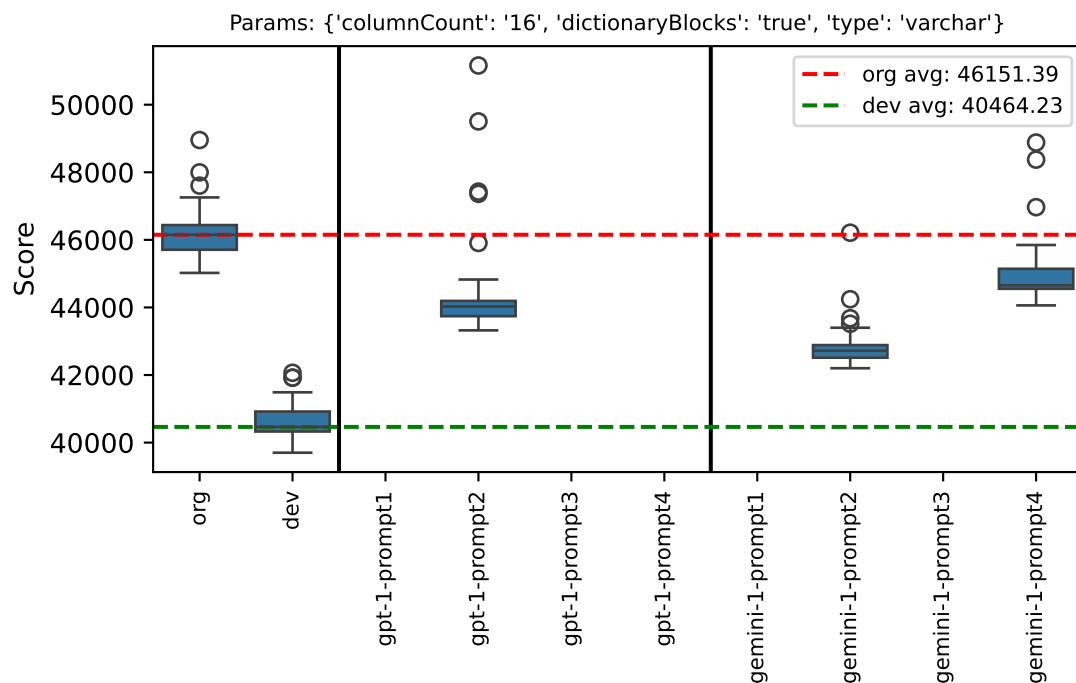
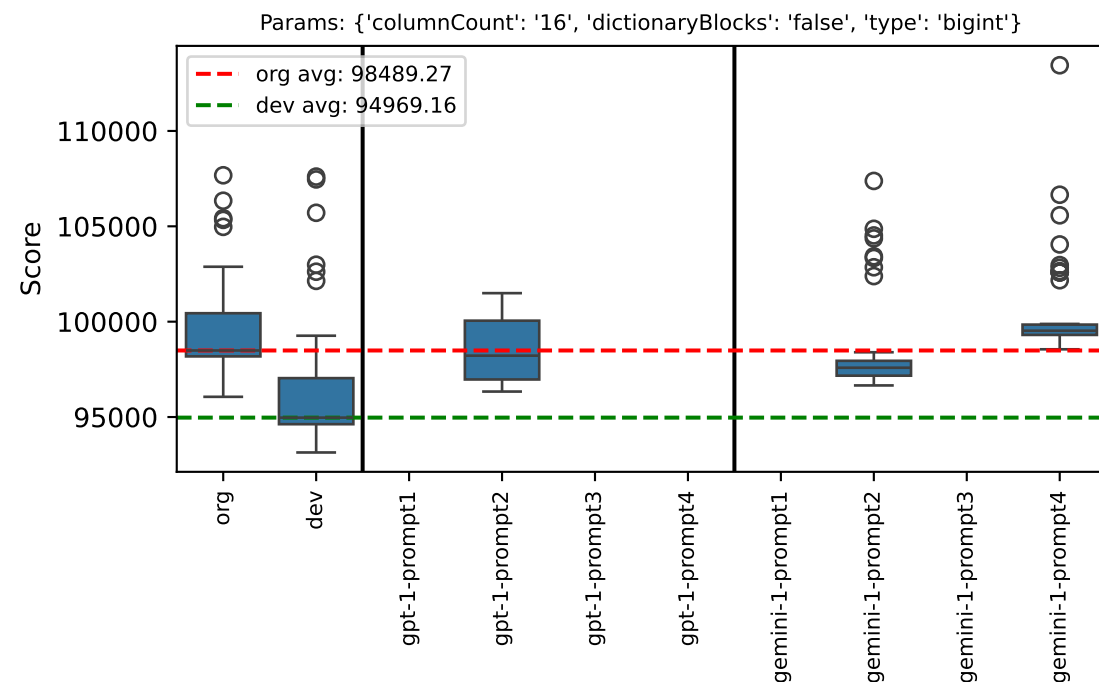
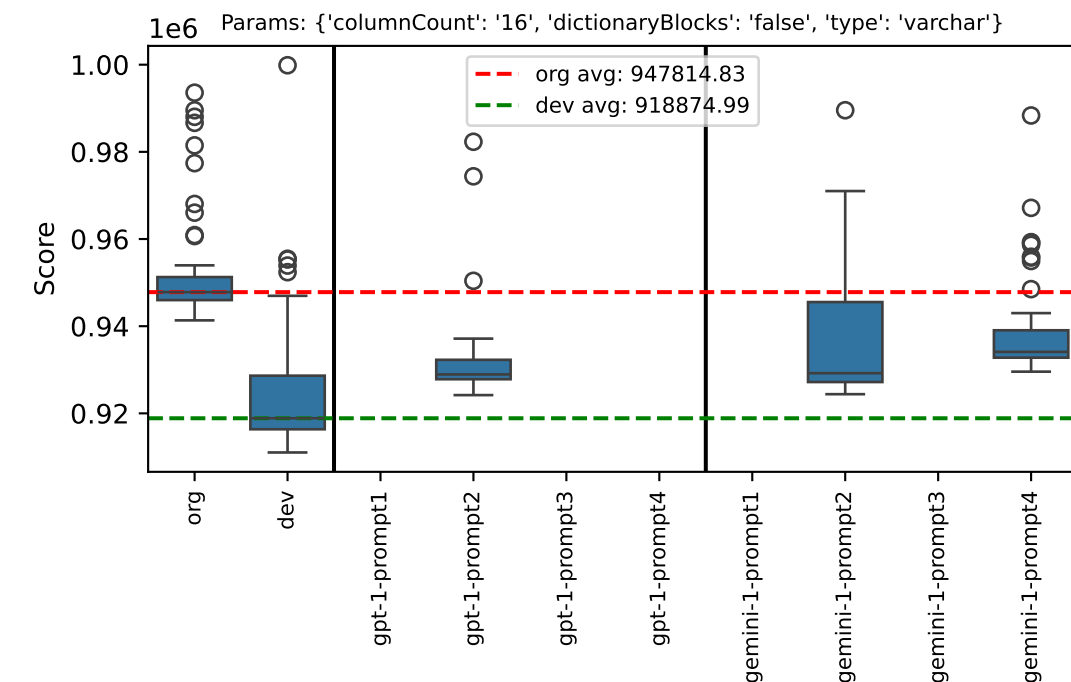
Params: {'columnCount': '8', 'dictionaryBlocks': 'true', 'type': 'varchar'}



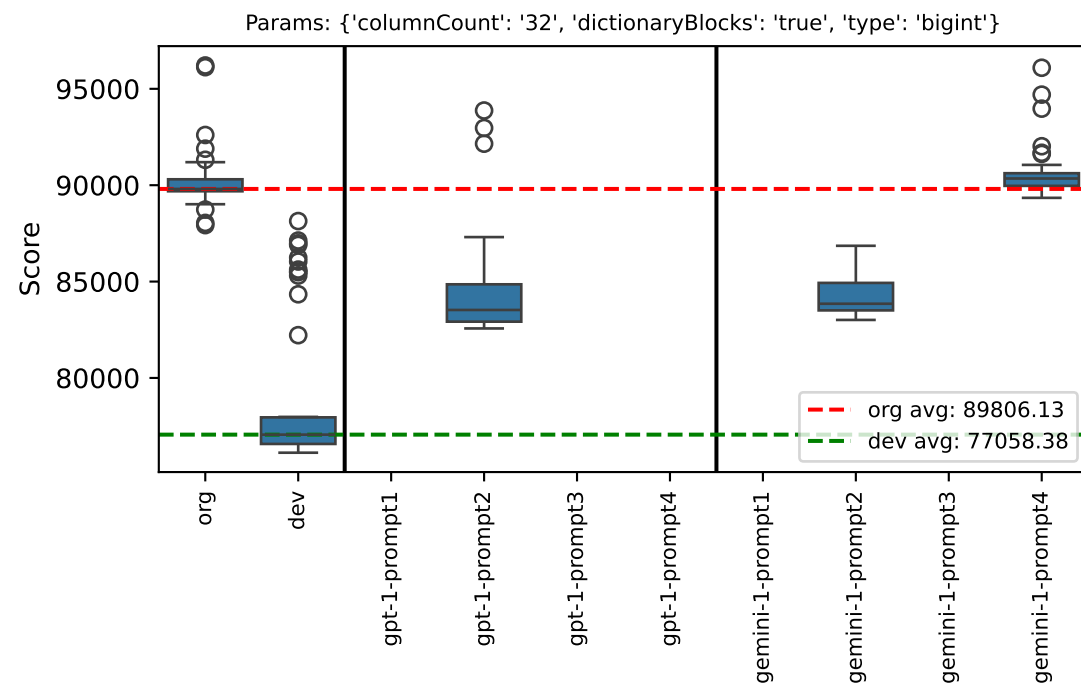
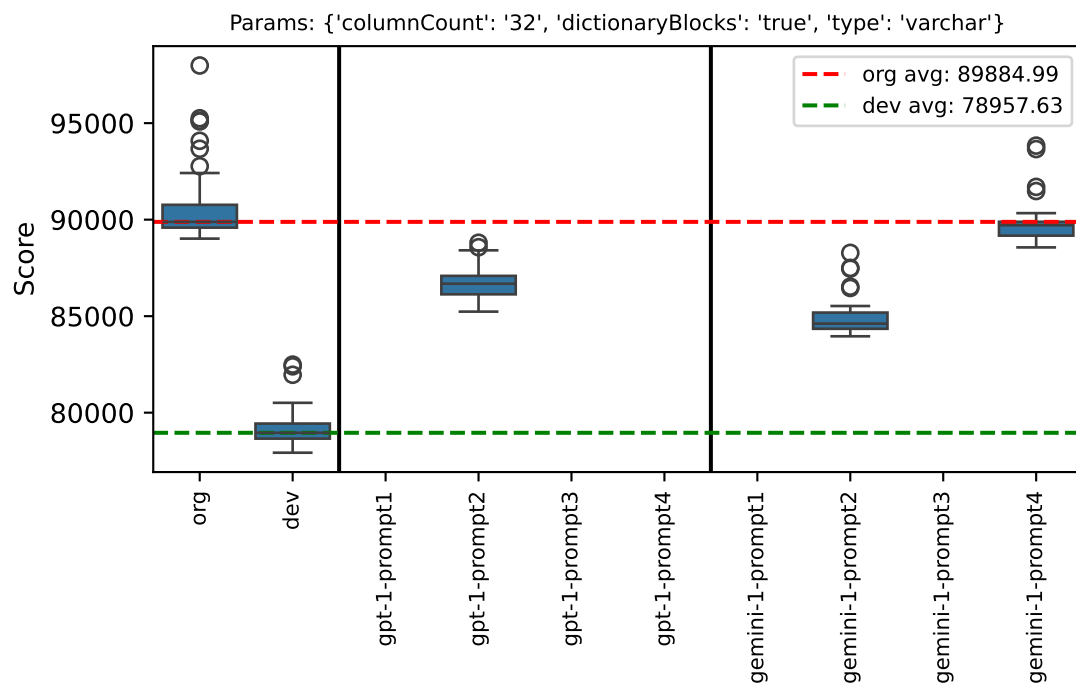
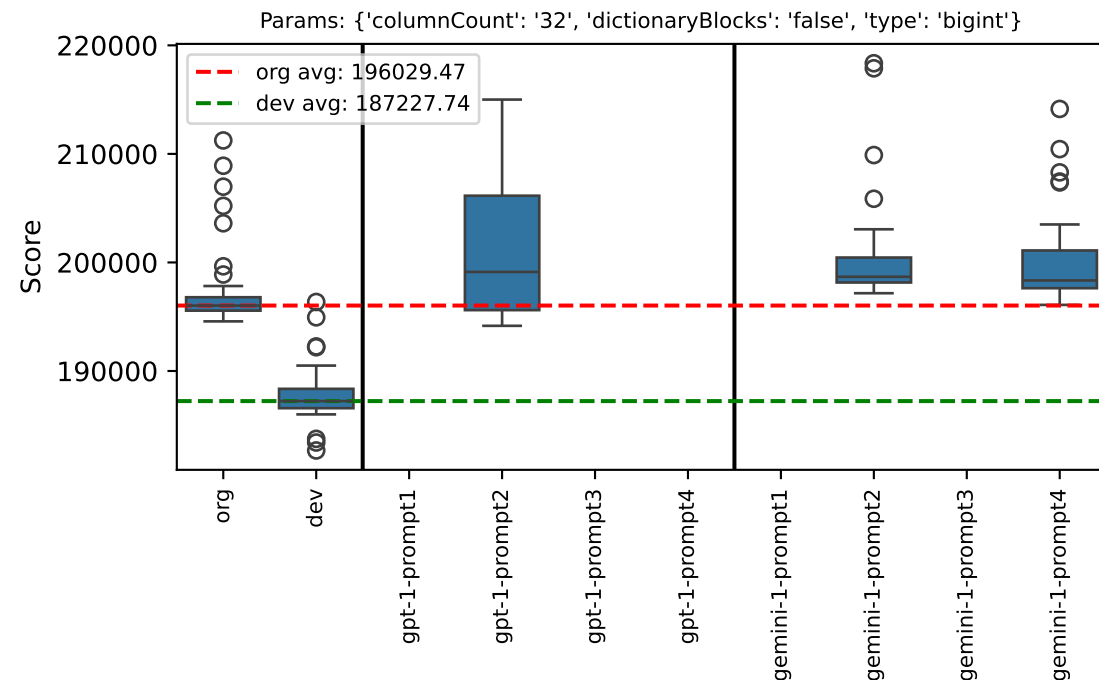
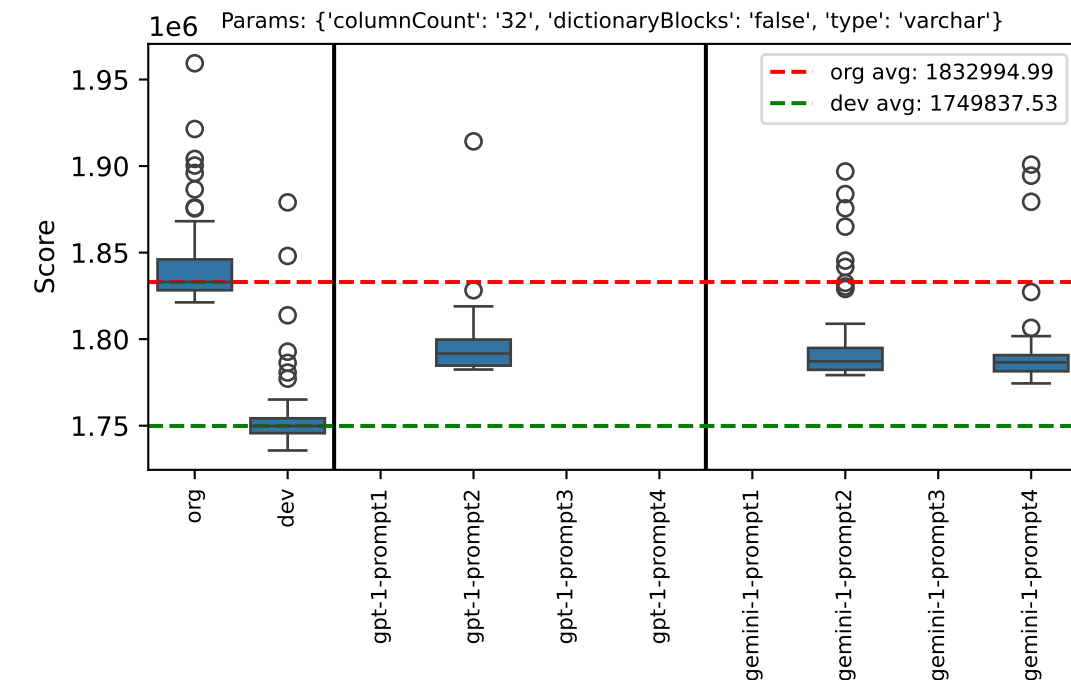
Params: {'columnCount': '8', 'dictionaryBlocks': 'true', 'type': 'bigint'}



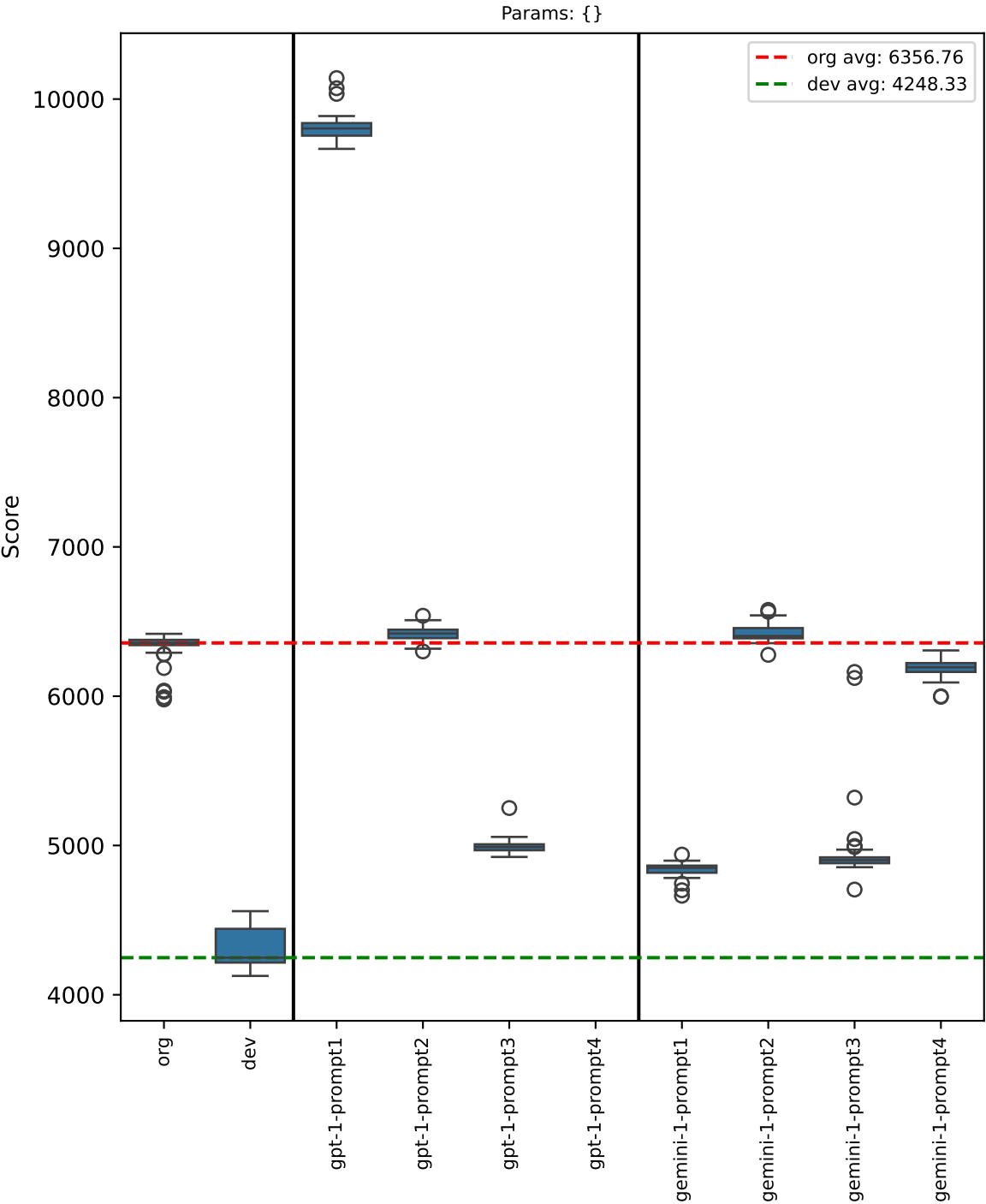
Benchmark: com.facebook.presto.sql.gen.PageProcessorBenchmark.columnOriented



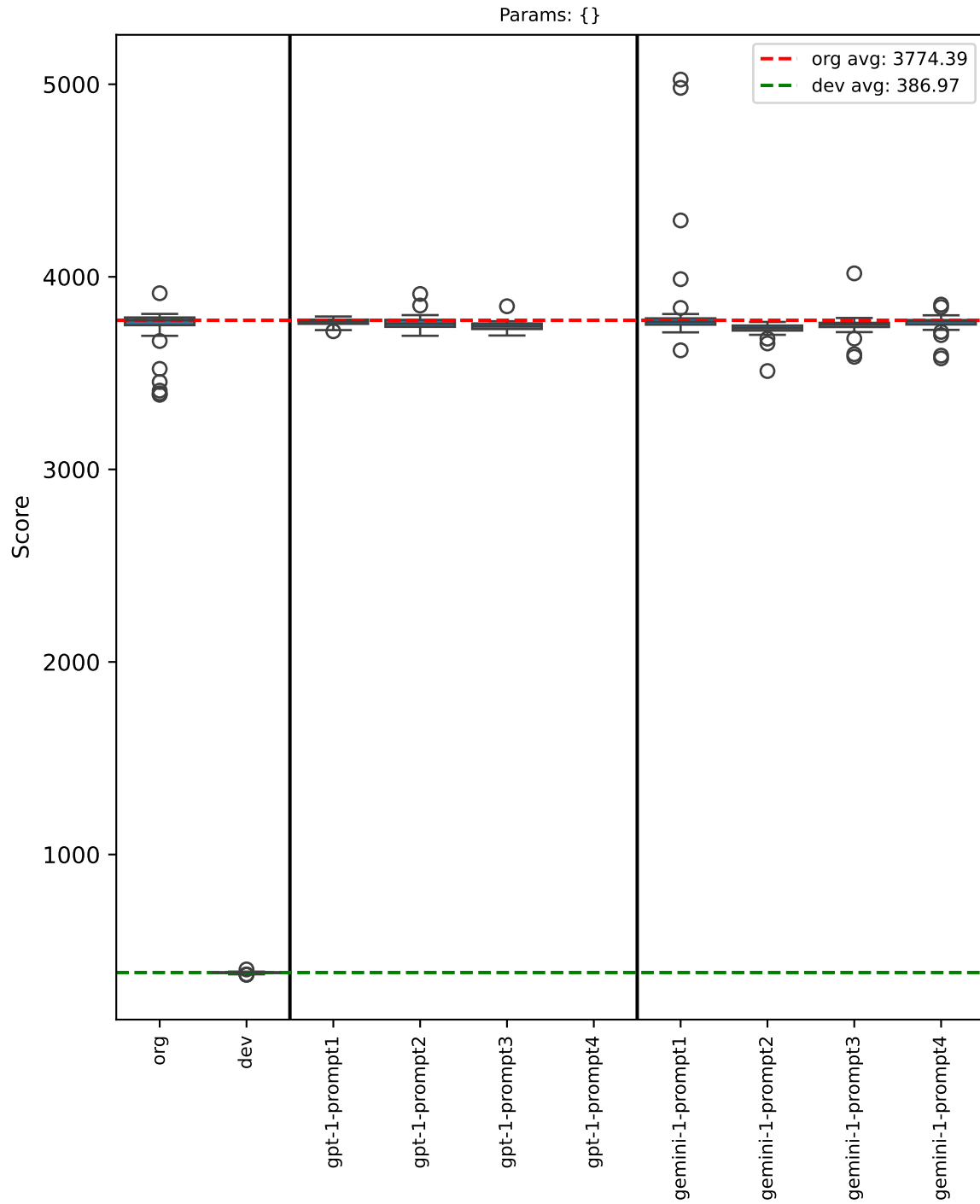
Benchmark: com.facebook.presto.sql.gen.PageProcessorBenchmark.columnOriented



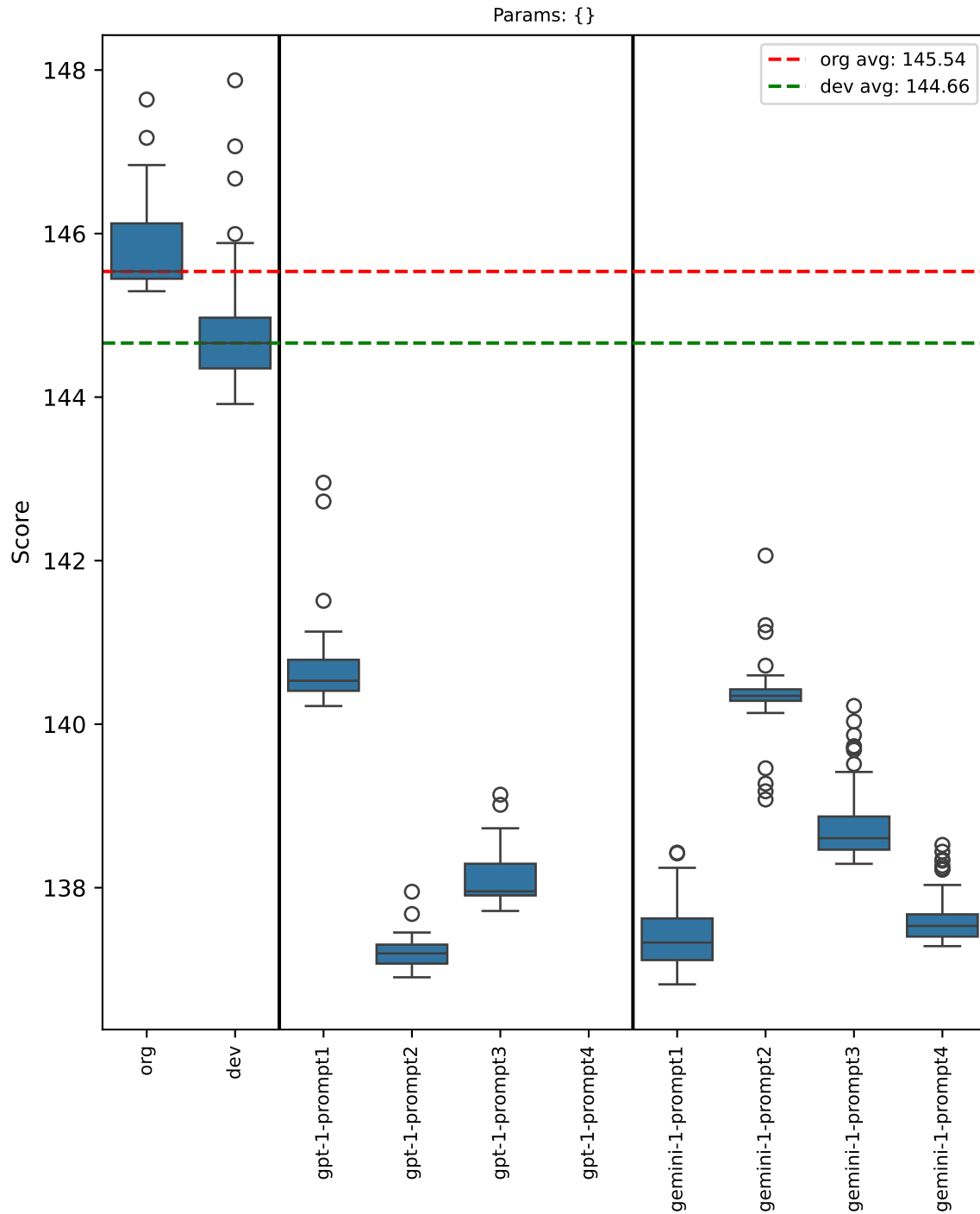
Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkCopyT



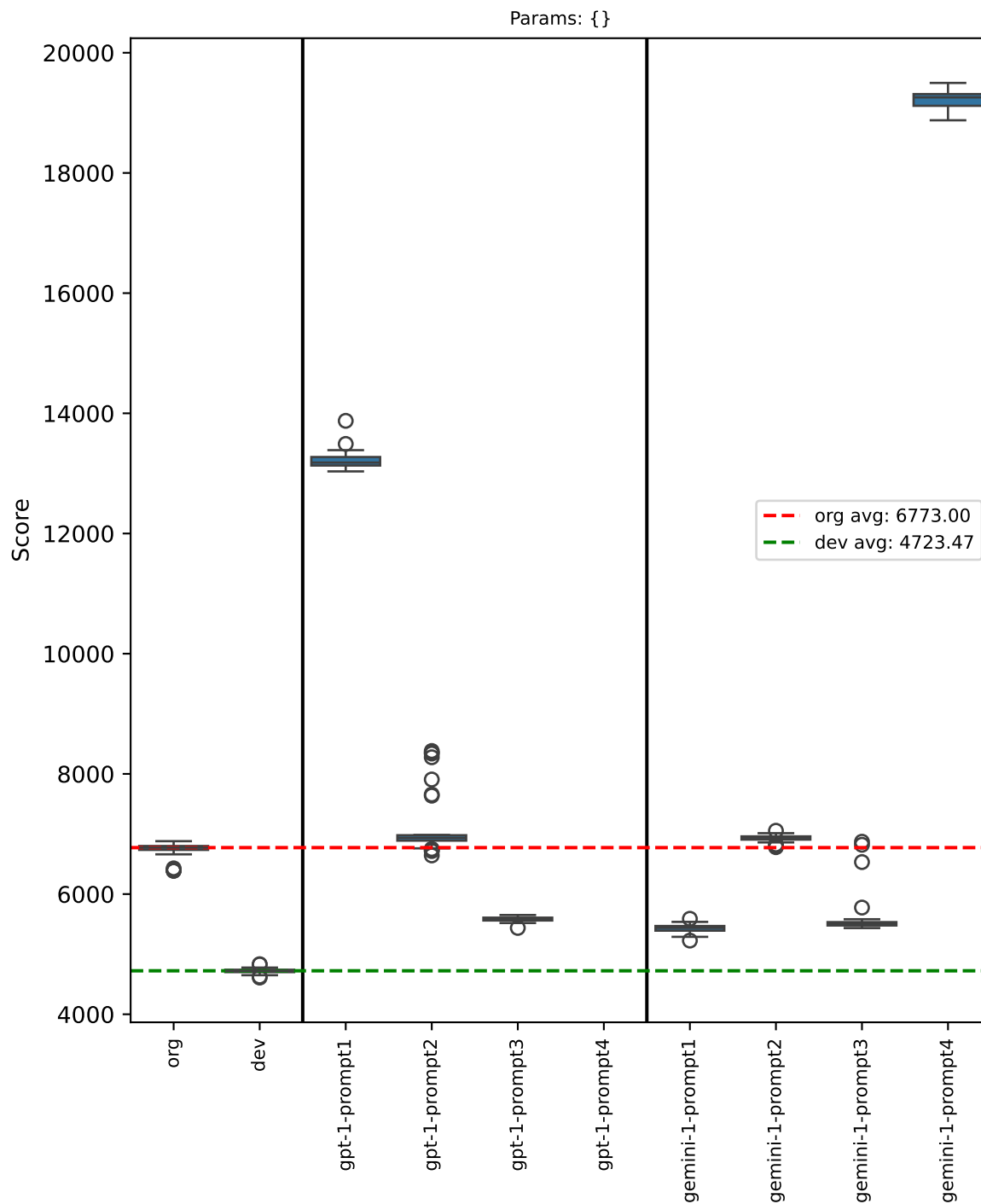
Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkDeserializeT



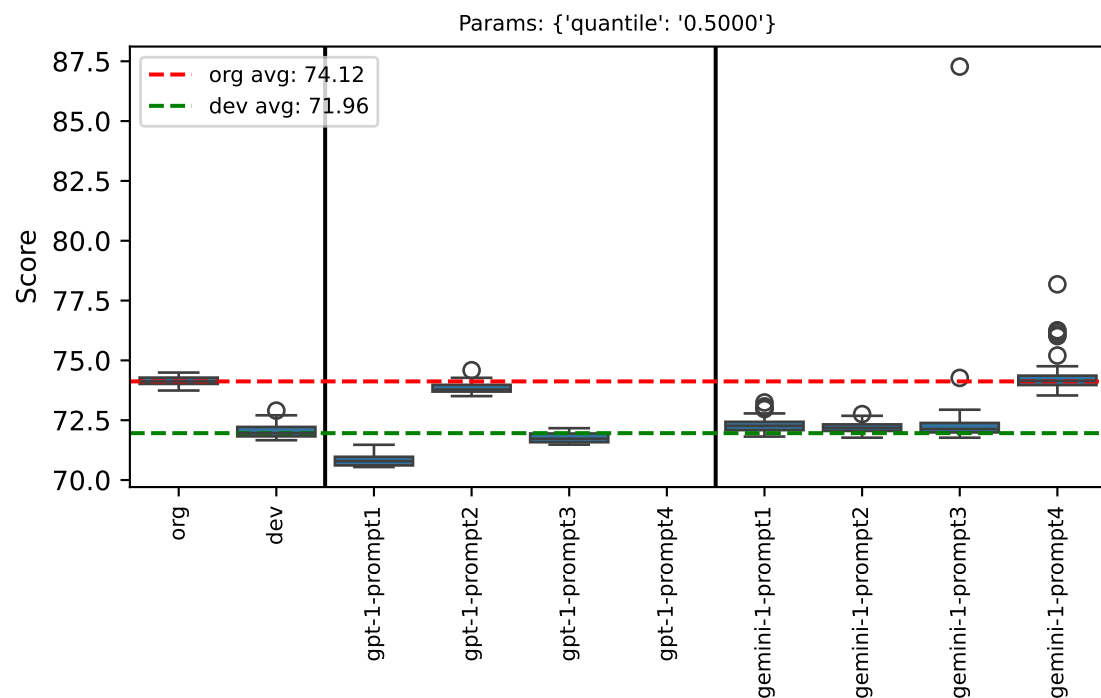
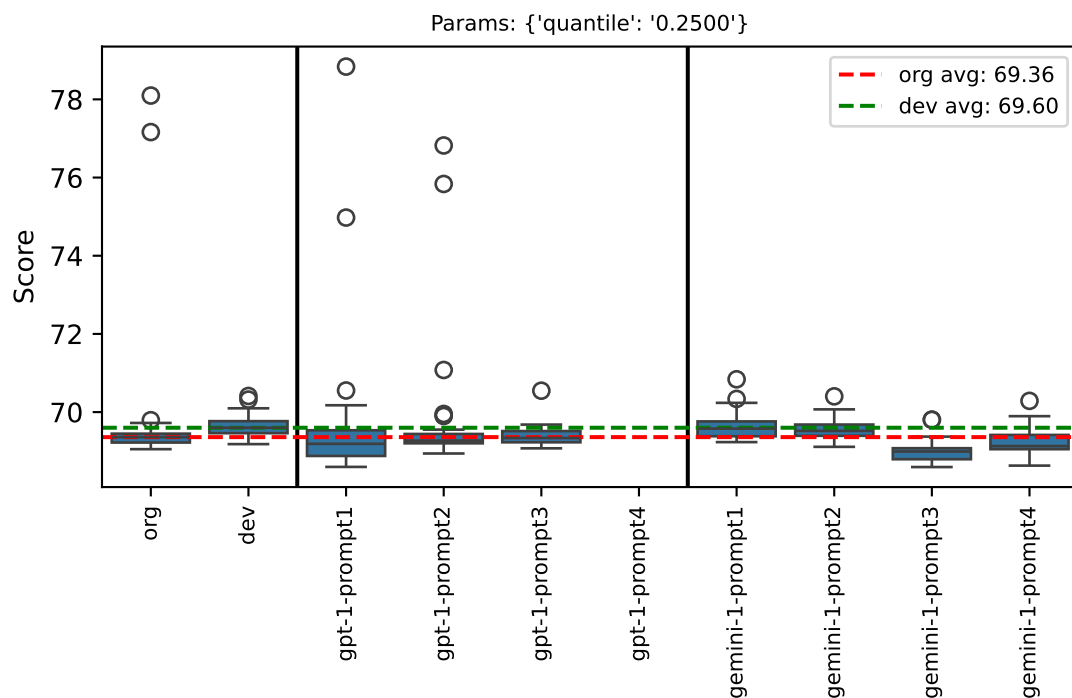
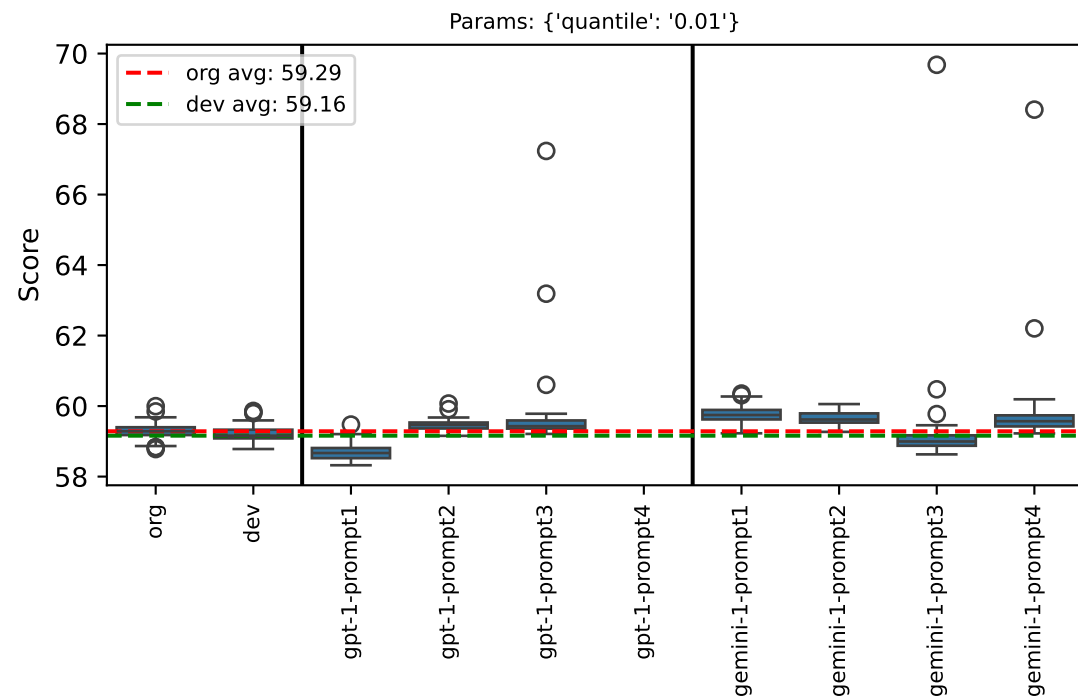
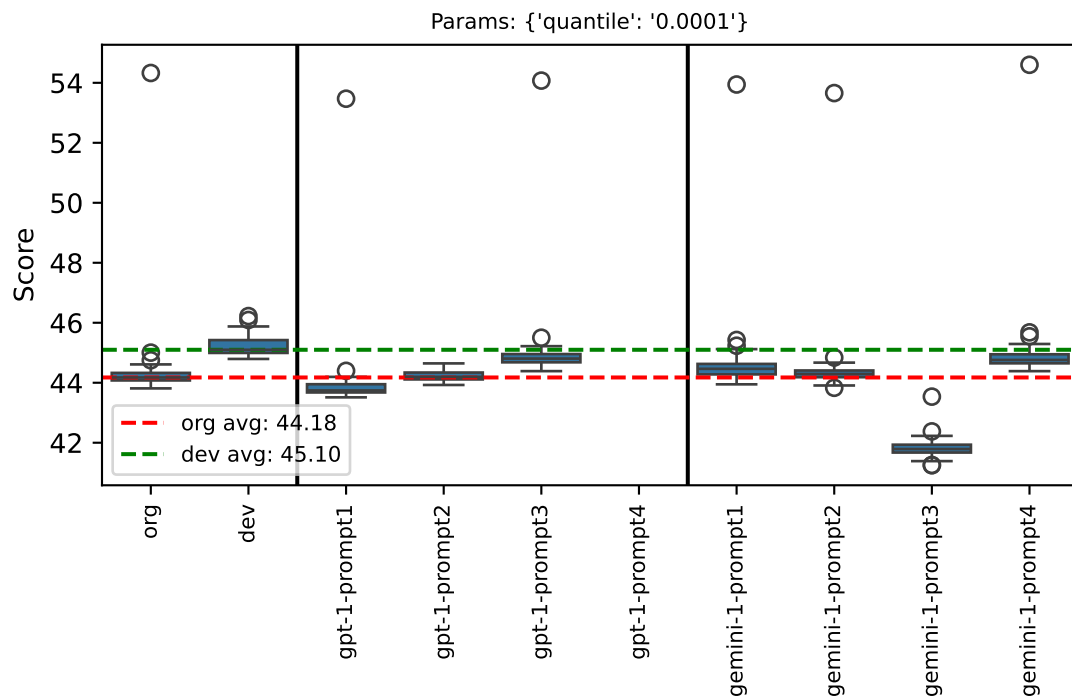
Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkInsertsT



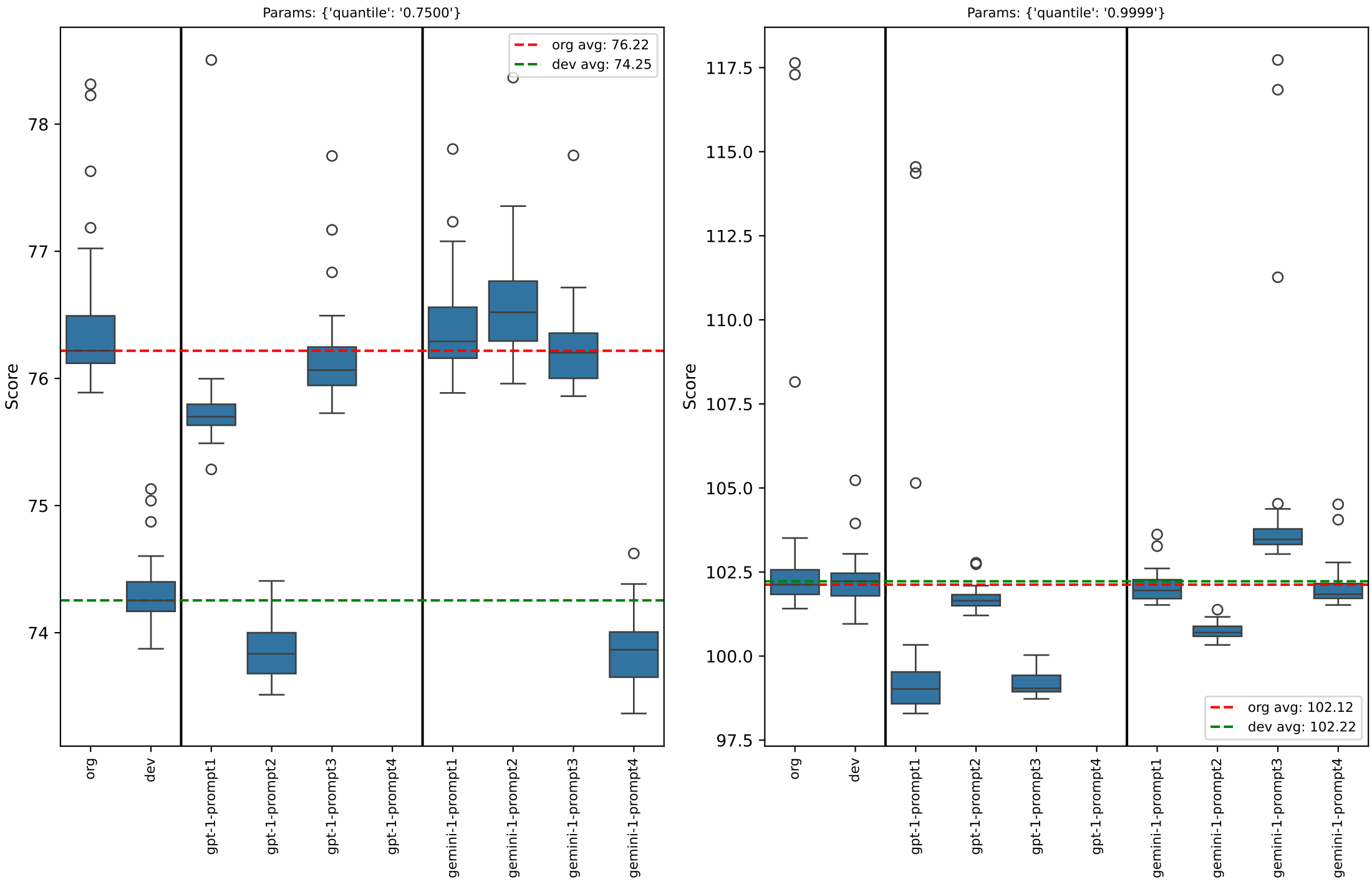
Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkMergeT



Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkQuantilesT



Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkQuantilesT



Benchmark: com.facebook.presto.tdigest.BenchmarkTDigest.benchmarkSerializeT

